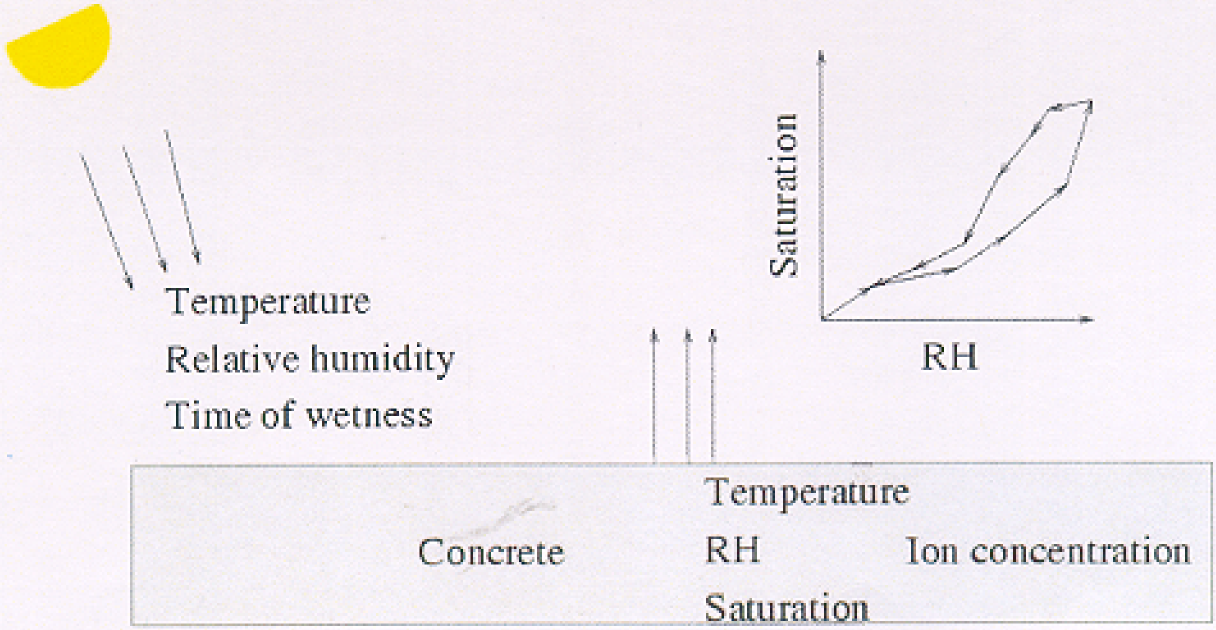
NISTIR 6551

A Computer Model to Predict the Surface Temperature and Time-of-Wetness of Concrete Pavements and Bridge Decks

Dale P. Bentz



1901-2001
NIST CENTENNIAL

NIST
National Institute of Standards and Technology
Technology Administration, U.S. Department of Commerce

NISTIR 6551

A Computer Model to Predict the Surface Temperature and Time-of-Wetness of Concrete Pavements and Bridge Decks

Dale P. Bentz
Building and Fire Research Laboratory

August 2000

U.S. Department of Commerce
Norman Y. Mineta, Secretary

Technology Administration
Dr. Cheryl L. Shavers, Under Secretary of Commerce for Technology

National Institute of Standards and Technology
Raymond G. Kammer, Director

ABSTRACT

This report provides documentation and example results for a computer model developed to predict the surface temperature and time-of-wetness (and time-of-freezing) of concrete pavements and bridge decks. The model is based on a one-dimensional finite difference scheme and includes heat transfer by conduction, convection, and radiation. Environmental conditions are varied by using the typical meteorological year datafiles available from the National Renewable Energy Laboratory. Based on the weather data available in these files and the developed heat transfer model, both the surface temperature and the time-of-wetness of the concrete are computed for typical pavement and bridge deck structures. The time-of-wetness includes both precipitation and condensation on the concrete surface. These predictions will be utilized as input into sorptivity-based service life models for sulfate attack and freeze/thaw deterioration, which is the main goal of the NIST research project being funded by the Federal Highway Administration.

Keywords: Bridge decks, building technology, computer modeling, concrete, heat transfer, pavements, surface temperature, time-of-wetness.

Contents

List of Figures

List of Tables

1 Introduction

As part of a research project being funded by the Federal Highway Administration (FHWA), the National Institute of Standards and Technology (NIST) is developing a computer program to predict the service life of concrete pavements and bridge decks exposed to sulfate attack and/or freeze-thaw conditions [1]. For these models, the major mechanism of transport of water and sulfate ions being considered is sorption by the partially saturated concrete. The uptake of water and the sulfate ions contained within it is dependent on many variables including the time-of-wetness of the concrete surface, the saturation of the concrete, and its surface temperature. The objective of this report is to describe the details of a computer model for predicting these three variables as a function of the environmental exposure for hardened concrete. Models for the thermal properties and temperature distribution of fresh hardening concrete have been presented in the literature previously [2, 3].

The developed computer model (CONCTEMP) is based on a one-dimensional finite difference scheme [4] and includes heat transfer due to conduction, convection, and radiation. The top surface boundary conditions (and the lower surface boundary conditions for bridge decks) are established based on the typical meteorological years weather data (TMY2DATA) provided by the National Renewable Energy Laboratory (NREL) [5]. Material thermal properties and convection coefficients are taken from the available literature. Outputs from the model include the variation of concrete surface temperature during the course of a year, a file detailing all time-of-wetness events (start time, duration, concrete surface temperature, and exterior relative humidity prior to event), and a file detailing time-of-freezing events (start time, duration, and minimum surface temperature achieved during freezing). These files have been created for twelve representative geographical locations throughout the United States. A complete listing of the computer code used to estimate the concrete surface temperature for pavement structures can be found in Appendix A. The program has been written in the C programming language. It has been written in a modular fashion with separate subroutines for reading in the input material properties and for updating the temperatures of all nodes within the 1-D structure. The log files for wetting and freezing events are created within the main program routine, which is also responsible for accumulating time throughout the year and reading in the hourly weather data values.

DISCLAIMER

This software was developed at the National Institute of Standards and Technology by employees of the Federal Government in the coursse of their official duties. Pursuant to title 17 Section 105 of the United States Code this software is not subject to copyright protection and is in the public domain. CONCTEMP is an experimental system. NIST assumes no responsibility whatsoever for its use by other parties, and makes no guarantees, expressed or implied, about its quality, reliability, or any other characteristic. We would appreciate acknowledgement if the software is used.

The U.S. Department of Commerce makes no warranty, expressed or implied, to users of CONCTEMP and associated computer programs, and accepts no responsibility for its use. Users of CONCTEMP assume sole responsibility under Federal law for determining the appropriateness of its use in any particular application; for any conclusions drawn from the results of its use; and for any actions taken or not taken as a result of analyses performed using these tools.

Users are warned that CONCTEMP is intended for use only by those competent in the field of cement-based materials and is intended only to supplement the informed judgment of the qualified user. The software package is a computer model which may or may not have predictive value when applied to a specific set of factual circumstances. Lack of accurate predictions by the model could lead to erroreous conclusions with regard to materials selection and design. All results should be evaluated by an informed user.

INTENT AND USE

The algorithms, procedures, and computer programs described in this report constitute a methodology for predicting the surface temperature and time-of-wetness of concrete under typical weathering conditions. They have been compiled from the best knowledge and understanding currently available, but have important limitations that must be understood and considered by the user. The program is intended for use by persons competent in the field of cement-based materials and with some familiarity with computers. It is intended as an aid in the materials selection, optimization, and design process.

2 Heat Transfer Model

The basic geometrical configurations for concrete pavements and bridge decks considered in this study are shown in Figure 1. For a pavement, the concrete thickness is considered to be 0.2 m, with a 0.2 m thick layer of soil (subbase) immediately beneath it. For the bridge deck, both surfaces of the concrete are assumed to be exposed to the environment, which is generally the case when temporary wooden forms are used [6, 7]. While often the bottom surface is protected from the environment by steel forms which remain in place [6, 7], the thickness of these forms (20 gauge or 0.9 mm) is such that their contribution to heat transfer should be minimal. For the bridge decks, it is further assumed that the convection coefficient for heat transfer is the same for both the top and bottom surfaces, and that no heat transfer by radiation (incoming sunlight or emitted radiation) occurs at the bottom surface. For the pavements, it is further assumed that at a depth of 0.2 m, the soil temperature is constant at a value of 13 °C. (Variation of this soil temperature did not significantly influence the model results in this study).

Both systems shown in Figure 1 are modelled using a one-dimensional finite difference grid with a spatial resolution of 20 mm (e.g., ten nodes within the concrete). At the top surface, four modes of heat transfer are considered: conduction into the concrete, convection, solar absorption, and grey-body irradiation to the surroundings. Specifically, the heat flow contribution (in $\frac{W}{m^2}$) due to conduction is given by:

$$Q_{cond} = k_{conc} \times \frac{(T_0 - T_1)}{\Delta x} \tag{1}$$

where k_{conc} is the thermal conductivity of the concrete in $\frac{W}{m\ ^\circ C}$, T_0 and T_1 are the surface temperature and internal temperature at the first internal node, respectively, and Δx is the node spacing (20 mm). For convection, the heat flow is given by:

$$Q_{conv} = h_{conv} \times (T_0 - T_{ambient}) \tag{2}$$

where $T_{ambient}$ is the ambient temperature and h_{conv} is the convection coefficient in $\frac{W}{m^2\ ^\circ C}$. While several empirical relationships to estimate convection coefficients are available in the

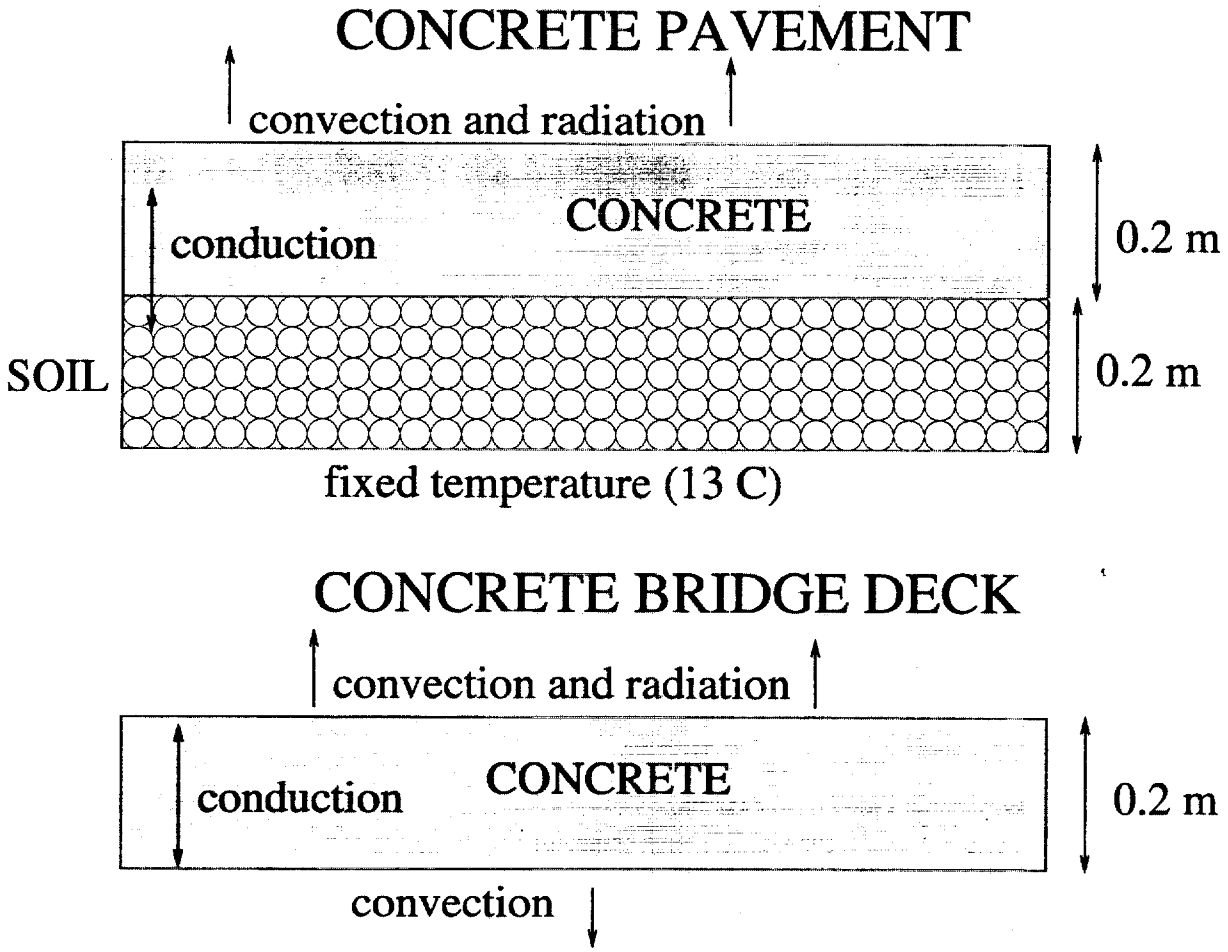


Figure 1: Basic configurations of one-dimensional heat transfer model for concrete pavements and bridge decks.

literature [2], for this study, the convection coefficient was calculated based on the wind speed available in the weather data files and the following equations used in the commercially available FEMMASSE system [8]:

$$h_{conv} = 5.6 + 4.0 v_{wind} \qquad \text{for } v_{wind} \leq 5 \ m/s \tag{3}$$

$$h_{conv} = 7.2 \times v_{wind}^{0.78} \qquad \text{for } v_{wind} > 5 \ m/s$$

where v_{wind} is the measured wind speed in m/s.

For radiative heat transfer at the top concrete surface, two contributions are considered. The first is the radiation absorbed from the incoming sunlight. A simple equation for the incoming heat flow due to this source is given by [2]:

$$Q_{sun} = \gamma_{abs} * Q_{inc} \tag{4}$$

where Q_{inc} is the incident solar radiation (W/m^2) and γ_{abs} is the solar absorptivity of the concrete. For this study, the incident solar radiation was taken directly from the weather data files [5], while a value of 0.65 was used for the solar absorptivity [8]. Second, one must consider the emission of radiation from the "warm" concrete to the night sky. This heat flow can be estimated as:

$$Q_{sky} = \sigma\epsilon \times (T_{0K}^4 - T_{sky}^4) \tag{5}$$

where σ is the Stefan-Boltzmann constant (5.669×10^{-8} W/(m^2 °C^4)), ϵ is the emissivity of the concrete, T_{0K} is the concrete surface temperature (in K), and T_{sky} is the calculated sky temperature also in K. For this study, a value of 0.9 was used for the concrete emissivity [2, 8, 9]. The sky temperature was estimated based on an algorithm presented by Walton [10] using the following series of equations:

$$T_{sky} = {\epsilon_s}^{\frac{1}{4}} \times T_{ambient} \quad (T\ in\ K) \tag{6}$$

where the sky emissivity, ϵ_s is given by:

$$\epsilon_s = 0.787 + 0.764 \times ln(\frac{T_{dew}}{273.}) \times F_{cloud} \tag{7}$$

where T_{dew} is the dewpoint temperature in K and with the cloud cover factor, F_{cloud}, as:

$$F_{cloud} = 1.0 + 0.024N - 0.0035N^2 + 0.00028N^3 \tag{8}$$

where N is the "tenths cloud cover", taking values between 0.0 and 1.0. Because the above equations for surface irradiation are significantly different from approximations used elsewhere [2], it is worth noting that the equations provided here have been successfully used in a variety of predictive models of relevance to the construction community [10, 11].

The above equations are employed in a general finite difference solution to one-dimensional heat transfer [4]. The time step in the finite difference scheme is established based on the layer thickness (20 mm) and the thermal properties of the materials, to ensure numerical convergence of the solution [4]. The ambient environmental conditions at any specific time are calculated by linear interpolation of the hourly values available from the weather files.

2.1 Material Properties

The additional material properties needed for the heat transfer model are provided in Table 1. The values for the soil were taken directly from the Final Report for the FHWA HIPERPAV project [2]. For concrete, a variety of values for the properties are available in the literature [2, 3, 4, 12] and the values provided in Table 1 are considered to be reasonable "mean" estimates. Of course, it is well recognized that each of these material properties will vary with the specific concrete under consideration. If specific values were measured for a concrete of interest, the heat transfer model could be easily executed for these values, since they are typically provided in an input file to the main program.

2.2 Environmental Conditions

When considering the service life of a field concrete, the environmental conditions to which the concrete is exposed are equally as important as its transport and mechanical properties. Naturally, these conditions are highly variable thoughout the United States. For the current study, realistic environmental exposures were obtained by using the typical meteorological year weather data files available from NREL [5]. These data files provide typical hourly

Table 1: Material Properties for Heat Transfer Model

Material	Heat Capacity C_p $(\frac{J}{kg\ ^\circ C})$	Thermal Conductivity k $(\frac{W}{m\ ^\circ C})$	Density ρ $(\frac{kg}{m^3})$
Concrete	1000	1.5	2350
Soil	800	0.3	1600

weather data for over 200 geographical locations within the United States, each geographical location being characterized by a five digit numerical code. For the heat transfer model described above, the following TMY2DATA parameters are utilized: ambient temperature ($T_{ambient}$), dewpoint temperature (T_{dew}), ambient relative humidity, wind speed (v_{wind}), precipitation, cloud cover (N), and incident global horizontal radiation (Q_{inc}).

As the heat transfer model executes, two types of events are logged, namely wetting and freezing. A wetting event may be due either to a precipitation event (rain, drizzle, sleet, or hail) in the weather data file or condensation when the computed surface temperature of the concrete falls below the current dewpoint temperature (but is still above 0 °C). It should be noted that snow is not considered a wetting event as we are assuming that the snow will be successfully removed from the pavement or bridge deck before substantial melting/wetting occurs. A freezing event is defined by the concrete surface temperature falling below 0 °C. This value was chosen as a conservative estimate, since the freezing point of pore water in concrete is typically depressed by the presence of ionic species and the small sizes of the capillary and gel pores in the concrete. When a wetting event occurs, the following values are logged to a file: the initiation time (hours since the beginning of the year), the duration of the event in hours, the average of the concrete surface temperatures at the initiation and the termination of the event, and the ambient relative humidity at the initiation of the event. For a freezing event, only the initiation time, duration, and minimum achieved surface temperature are logged to a freezing event data file.

3 Example Results

3.1 Concrete Surface Temperature vs. Time of Year

Three examples of predicted concrete surface temperatures as a function of geographical location and season of the year are provided in Figures 2, 3, and 4. Little data is available in the literature to validate these predictions, but the data of both Andrade [13] and Basheer and Nolan [14] support the general trends shown in the three figures. Specifically, during the day, the concrete surface temperature generally rises above the ambient temperature due to the incoming solar radiation. At night, the concrete temperature falls due to radiation from the concrete surface to the sky, sometimes falling below the ambient air temperature and occasionally falling below the dewpoint. This effect is more significant for bridge decks than for pavements since there is no soil sublayer providing a thermal mass beneath the concrete in the case of bridge decks.

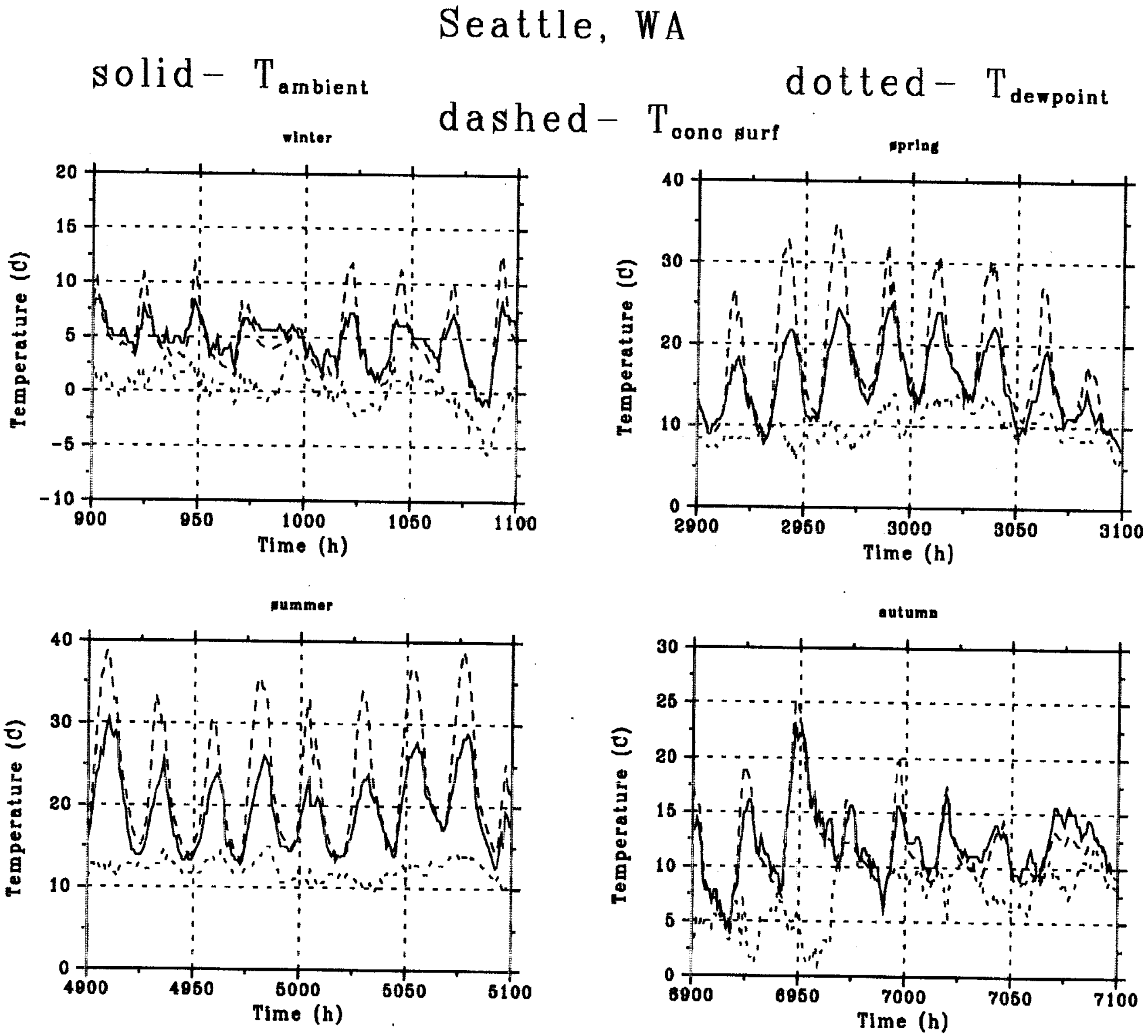


Figure 2: Temperature predictions for a concrete pavement surface in Seattle, Washington.

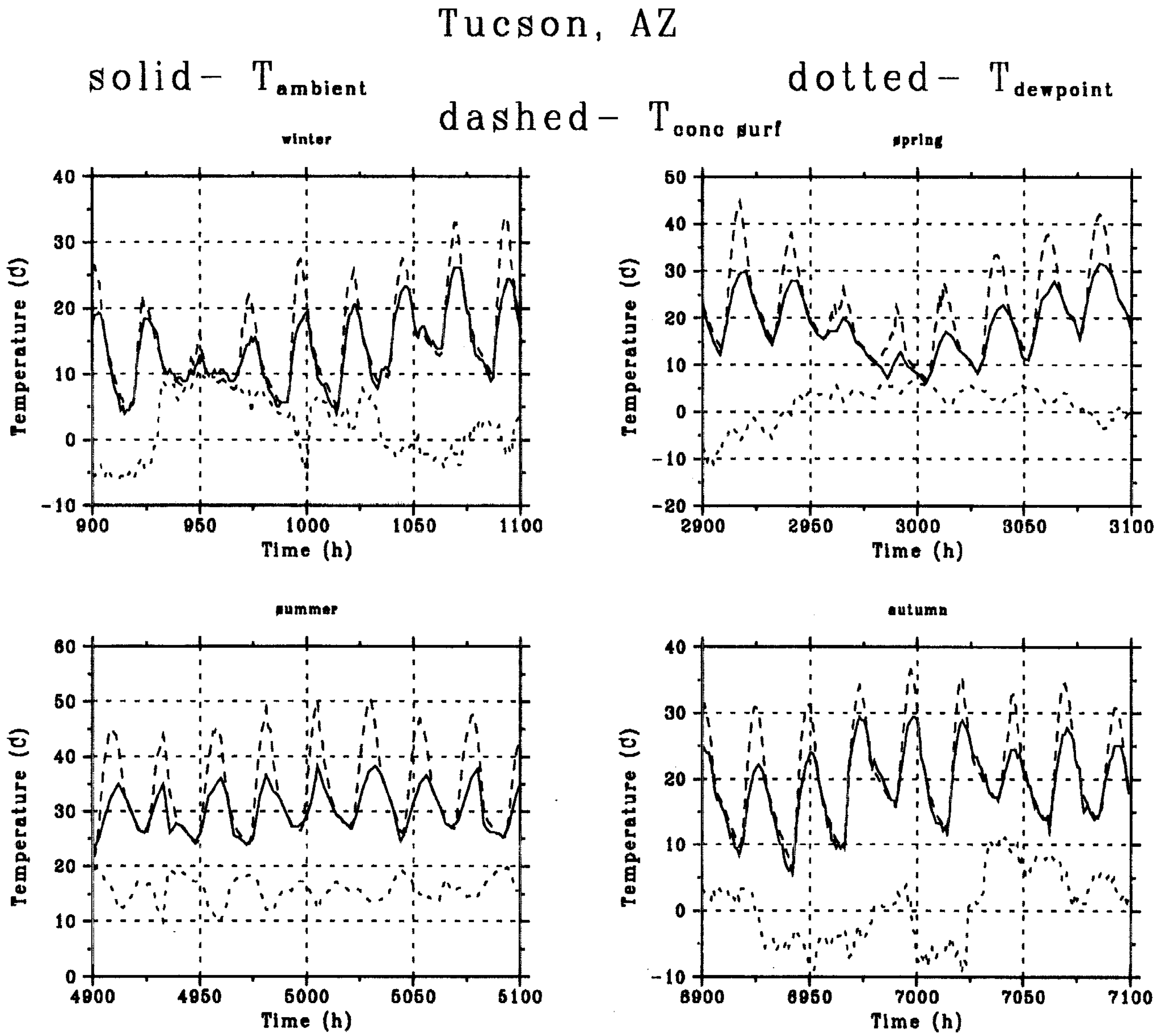


Figure 3: Temperature predictions for a concrete pavement surface in Tucson, Arizona.

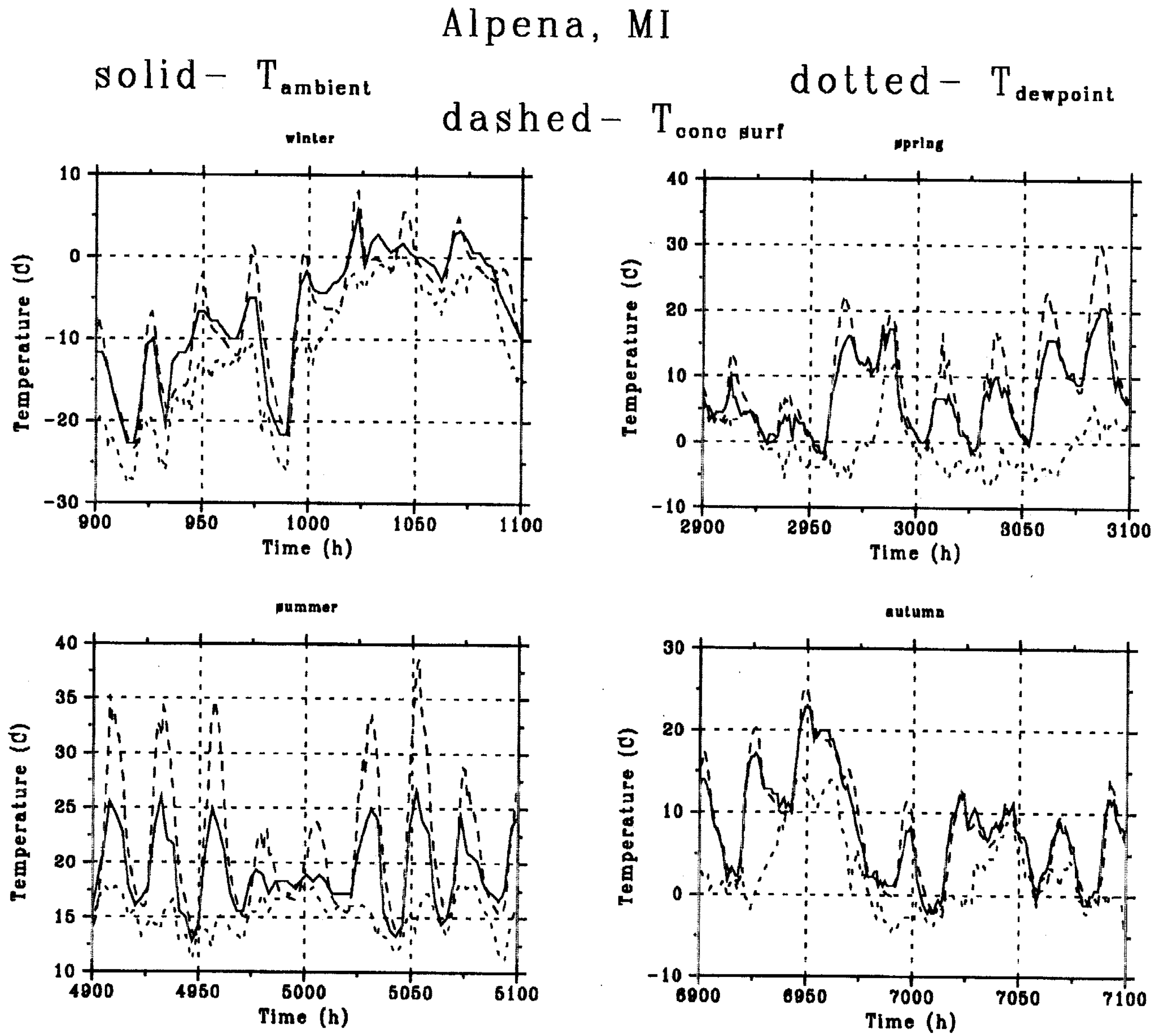


Figure 4: Temperature predictions for a concrete bridge deck surface in Alpena, Michigan.

3.2 Time-of-Wetness and Time-of-Freezing Predictions

The wetting and freezing events for 12 representative geographical locations are summarized in Tables 2 and 3. It is easily observed that there is considerable variability in time-of-wetness and number of freeze-thaw cycles across the United States. Of the 12 locations investigated to date, Tucson, AZ is the driest with the concrete pavement being wet only 128 h per year. Conversely, Seattle, WA is the wettest climate with the pavement being wet 1698 h per year. Obviously, the time-of-wetness and number of freeze-thaw cycles will have a significant influence on the durability of concrete pavements and bridge decks in any given geographical location. This influence will be further developed in the sorptivity-based service life models for sulfate attack and freeze-thaw deterioration [1] currently being developed as one of the final stages of this research project.

For freezing events, there are always more freeze-thaw cycles for the bridge decks than for the pavements (in agreement with the ubiquitous posting of road signs indicating that "bridge freezes before roadway"). Not surprisingly, Tampa, FL experiences the fewest freeze-thaw cycles, with only the bridge decks experiencing any cycles in a representative year. Cheyenne, WY experiences the most freeze-thaw cycles, with over 125 cycles for both the pavements and bridge decks being observed in a representative year. In comparing bridge decks to pavements, Baltimore, MD exhibits the greatest difference, with the bridge deck concrete experiencing 21 more freeze-thaw cycles per year than a comparable pavement. This is most likely due to the somewhat moderate winter climate in Baltimore, with the ambient often near 0 °C, so that the bridge decks often experience freezing while the nearby pavements are just above the freezing point.

Table 2: Wetting and Freezing Events for Concrete Pavements

U.S. City	NREL code	Number of wetting events	Total h/yr wet	Average RH before wetting	Number of F-T cycles
Kansas City, MO	03947	173	448	83.7	79
Tampa, FL	12842	244	613	86.0	0
Lubbock, TX	23042	138	348	83.6	61
Tucson, AZ	23160	78	128	66.3	9
Cheyenne, WY	24018	117	224	71.5	126
Pierre, SD	24025	117	190	85.3	92
Seattle, WA	24233	407	1698	81.6	25
Fresno, CA	93193	139	393	86.2	14
Baltimore, MD	93721	238	948	84.7	83
Bridgeport, CT	94702	229	835	87.2	90
Alpena, MI	94849	218	680	86.1	102
Waterloo, IA	94910	209	551	86.7	72

Table 3: Wetting and Freezing Events for Concrete Bridge Decks

U.S. City	NREL code	Number of wetting events	Total h/yr wet	Average RH before wetting	Number of F-T cycles
Kansas City, MO	03947	179	490	84.1	81
Tampa, FL	12842	268	639	86.8	4
Lubbock, TX	23042	148	413	84.3	71
Tucson, AZ	23160	78	141	66.3	16
Cheyenne, WY	24018	125	252	72.9	131
Pierre, SD	24025	123	229	85.7	100
Seattle, WA	24233	413	1812	81.7	34
Fresno, CA	93193	161	521	87.0	20
Baltimore, MD	93721	254	972	85.3	104
Bridgeport, CT	94702	229	860	87.0	104
Alpena, MI	94849	239	758	86.8	107
Waterloo, IA	94910	212	623	86.8	86

4 Summary

A simple one-dimensional heat transfer model has been applied to estimating the surface temperature of concrete pavements and bridge decks. Using representative weather data from the NREL, the results indicate considerable geographical variability not only in the concrete surface temperatures, but also in the time-of-wetness and number of freeze-thaw cycles. This indicates that realistic estimates of service life can only be made when detailed consideration is given to the environment specific to the field concrete in question. The results of this model will be utilized as input to a Windows-based computer program for sorptivity-based service life predictions for the cases of sulfate attack and freeze-thaw deterioration, currently under development at NIST.

5 Acknowledgements

This study is being funded by the Federal Highway Administration. The assistance of the contract office technical representatives, Dr. Stephen Forster and Marcia Simon, is greatly appreciated. The author would like to thank Dr. George Walton (NIST/BFRL), Mr. Douglas Burch (formerly of BFRL), and Dr. Erik Schlangen of Intron (The Netherlands), for useful discussions and suggestions.

References

[1] Bentz, D.P., Clifton, J.R., Ferraris, C.F., and Garboczi, E.J., *Transport Properties and Durability of Concrete: Literature Review and Research Plan*, NISTIR **6395**, U.S. Department of Commerce, September 1999.

[2] McCullough, B.F., and Rasmussen, R.O., *Fast-Track Paving: Concrete Temperature Control and Traffic Opening Criteria for Bonded Concrete Overlays, Volume 1: Final Report*, **FHWA-RD-98-167**, Federal Highway Administration, October 1999.

[3] Kapila, D., Falkowsky, J., and Plawsky, J.L., *ACI Materials Journal*, **94** (2), 119-128, 1997.

[4] Holman, J.P., Heat Transfer (McGraw-Hill Book Company, New York, 1981).

[5] Marion, W., and Urban, K., *User's Manual for TMY2s: Typical Meteorological Years*, National Renewable Energy Laboratory, June 1995.

[6] Lane, D.S., Virginia Department of Transportation, personal communication.

[7] Felag, M., Rhode Island Department of Transportation, personal communication.

[8] Schlangen, E., *Online Help/Manual module HEAT of FEMMASSE, 1990-2000*, FEMMASSE b.v., The Netherlands, 2000.

[9] Siegel, R., and Howell, J.R., Thermal Radiation Heat Transfer, 3rd. Ed. (Hemisphere Publishing Corporation, Washington, 198X).

[10] Walton, G.N., *Thermal Analysis Research Program- Reference Manual*, NBSIR **83-2655**, U.S. Department of Commerce, March 1983, Update 1985.

[11] Zarr, R.R., *Analytical Study of Residential Buildings with Reflective Roofs*, NISTIR **6228**, U.S. Department of Commerce, October 1998.

[12] De Schutter, G., and Taerwe, L., *Magazine of Concrete Research*, **47** (172), 203-208, 1995.

[13] Andrade, C., Sarria, J., and Alonso, C., *Cement and Concrete Research*, **29**, 1249-1259, 1999.

[14] Basheer, P.A.M., and Nolan, E., Near-surface Moisture Gradients and In-situ Permeation Tests, Proceedings of a Special Technical Session on "Near-Surface Testing for Strength and Durability of Concrete", Fifth CANMET/ACI International Conference on Durability of Concrete, 91-114, 2000.

Appendix

A Source Code Listing for Heat Transfer Model for Concrete Pavements

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

#define STEFAN 5.669e-8   /* Stefan-Boltzmann constant */
#define EMISS 0.90        /* Emissivity of concrete surface */
#define SOLARABS 0.65     /* Solar absorptivity  of concrete */

/***********************************************************************/
/*                                                                     */
/*      Program: Tempcalc.c                                            */
/*      Program to calculate concrete surface temperature during a     */
/*      yearly weather cycle (based on a 1-D finite difference solution */
/*                                                                     */
/*      Inputs: File inputs.par with material properties of concrete   */
/*              and soil.                                              */
/*              Weather data file XXXXX.tm2 from NREL database         */
/*      Outputs: Files with                                            */
/*              1) surface temperature history (temphist.out)          */
/*              2) time-of-wetness history (wettime.out)               */
/*              3) time-of-freezing history (freezetime.out)           */
/*                                                                     */
/*      Programmer: Dale P. Bentz                                      */
/*      Last revision: July 19, 2000                                   */
/*      Effort funded by the Federal Highway Administration            */
/*                                                                     */
/***********************************************************************/

/* Global variables */
        double tempnew[22],tempold[22],t_out=0.0,sum_t,tk_sky;
        double conc_heat_cap,conc_thermal_k,conc_density;
        double h_coeff=0.0,alpha_conc,t_cum=0.0;
        double soil_heat_cap,soil_thermal_k,soil_density,alpha_soil;

/* Routine to read in input property parameters for concrete and soil */
/* Called by: main program */
/* Calls: none */
read_inputs(){
        FILE *parfile;
```

```
        parfile=fopen("inputs.par","r");
        /* Note that all inputs are in SI units */
        fscanf(parfile,"%lf",&conc_heat_cap);
        printf("Concrete heat capacity is %lf J/kg/C\n",conc_heat_cap);
        fscanf(parfile,"%lf",&conc_thermal_k);
        printf("Concrete thermal conductivity is %lf W/m/C\n",conc_thermal_k);
        fscanf(parfile,"%lf",&conc_density);
        printf("Concrete density is %lf kg/m/m/m\n",conc_density);
        alpha_conc=conc_thermal_k/conc_heat_cap/conc_density;
        printf("Calculated concrete alpha value is %.12lf m*m/s\n",alpha_conc);
        fscanf(parfile,"%lf",&soil_heat_cap);
        printf("Soil heat capacity is %lf J/kg/C\n",soil_heat_cap);
        fscanf(parfile,"%lf",&soil_thermal_k);
        printf("Soil thermal conductivity is %lf W/m/C\n",soil_thermal_k);
        fscanf(parfile,"%lf",&soil_density);
        printf("Soil density is %lf kg/m/m/m\n",soil_density);
        alpha_soil=soil_thermal_k/soil_heat_cap/soil_density;
        printf("Calculated soil alpha value is %.12lf m*m/s\n",alpha_soil);
        fclose(parfile);
}

/* Routine to update all temperatures in the finite difference grid */
/* Called by: main program */
/* Calls: none */
upd_temp(q_in,t_amb,t_dew,cloud_cover,delta_x,delta_t)
        double q_in,t_amb,t_dew,cloud_cover,delta_x,delta_t;
        /*      q_in is in W/m*m
                t_amb and t_dew are in degrees Celsius
                cloud_cover is in tenths (0.1-1.0)
                delta_x is in meters
                delta_t is in seconds */
{
        int i;
        FILE *tempfile;
        double q_sun,q_sky,q_conv,q_cond,q_stor,q1,q2;
        double tk_amb,tk_dew,e_s,cloud_fact;

        tk_amb=t_amb+273.15;    /* Conversion to Kelvin */
        tk_dew=t_dew+273.15;    /* Conversion to Kelvin */
     /* Sky temperature computed based on algorithm of Walton et al. (TARP) */
        cloud_fact=1.0+0.024*cloud_cover-0.0035*cloud_cover*cloud_cover;
        cloud_fact+=0.00028*cloud_cover*cloud_cover*cloud_cover;
        /* Compute the sky emissivity e_s */
        e_s=(0.787+0.764*log(tk_dew/273.))*cloud_fact;
        /* Now estimate the sky temperature tk_sky */
        tk_sky=sqrt(sqrt(e_s))*tk_amb;
```

```
        /* Compute the four heat flow contributions at the top surface */
        q_sun=q_in*SOLARABS;
        q_sky=STEFAN*EMISS*(pow((tempold[0]+273.15),4.0)-pow(tk_sky,4.0));
        q_conv=h_coeff*(tempold[0]-t_amb);
        q_cond=(conc_thermal_k/delta_x)*(tempold[0]-tempold[1]);
        q_stor=q_sun-q_sky-q_conv-q_cond;

        /* Update the surface temperature */
    tempnew[0]=tempold[0]+q_stor*2.*delta_t/delta_x/conc_density/conc_heat_cap;

        /* First, the top concrete layer */
        /* Equations based on those presented in Heat Transfer by Holman */
        for(i=1;i<10;i++){
                tempnew[i]=((alpha_conc*delta_t)/(delta_x*delta_x))*
                  (tempold[i+1]+tempold[i-1])+(1.0-2.*alpha_conc*delta_t/
                   (delta_x*delta_x))*tempold[i];
        }

        /* Second, the interface between concrete and soil */
        q1=(conc_thermal_k/delta_x)*(tempold[10]-tempold[9]);
        q2=(soil_thermal_k/delta_x)*(tempold[10]-tempold[11]);
        tempnew[10]=tempold[10]-(q1+q2)*2.*delta_t/delta_x/
         (conc_density*conc_heat_cap+soil_density*soil_heat_cap);

        /* Last, the soil subbase */
        for(i=11;i<20;i++){
                tempnew[i]=((alpha_soil*delta_t)/(delta_x*delta_x))*
                  (tempold[i+1]+tempold[i-1])+(1.0-2.*alpha_soil*delta_t/
                    (delta_x*delta_x))*tempold[i];
        }
        /* Last soil node is constant at a temperature of 13 C */
        tempnew[20]=13.0;

/* Copy the new temperatures over the old ones to prepare for the next update */
        for(i=0;i<=20;i++){
                tempold[i]=tempnew[i];
        }
        /* Output temperature results once per hour */
        if((t_cum-t_out)>=3600.){
                tempfile=fopen("temphist.out","a");
                fprintf(tempfile,"%.2lf %.2lf %.2lf %.2lf ",
                        t_cum/3600.,t_amb,t_dew,tk_sky-273.15);
                t_out=t_cum;
                /* Output surface temperature */
                for(i=0;i<=1;i++){
                        fprintf(tempfile,"%lf ",tempnew[i]);
                }
```

```
        /* And temperature for every fifth node into concrete and soil */
        for(i=5;i<=20;i+=5){
            fprintf(tempfile,"%lf ",tempnew[i]);
        }
        fprintf(tempfile,"\n");
        fclose(tempfile);
    }
}

/* Main program to read in weather data file and call routines to */
/* set material properties and update temperatures */
/* Calls: read_inputs and upd_temp */
main(){
    FILE *infile,*wetfile,*frfile;
    char wba[80],city[80],state[80],latd[10],longd[10];
    char filein[80];
    register int i,j;
    int tzone,lat1,lat2,long1,long2,elev;
    int wetflag=0,duration,numcum=0,freezeflag=0;
    long int timeorg,timefin;
    long int timeforg,timeffin;
    int year,month,day,hour,exhr,exdnr,glhr,glhrflu,dnr,dnrflu;
    int dfhr,dfhrflu,glhi,glhiflu,dhi,dhiflu,dni,dniflu;
    int zi,ziflu,tskycov,tskycovflu,opqskycov,opqskycovflu,tempdry,tdflu;
    int tempwet,twflu,rh,rhflu,atmp,atmpflu,wdir,wdirflu,wspd,wspdflu;
    int vis,visflu,ceilflu,days[13];
    long int ceil,durcum;
    double qsun,temp_amb,temp_dew,del_x,del_t,del_t_soil,tfact;
    double temp_amb_old,wspd_old,temp_dew_old,wspd_new;
    double temp_amb_cur,wspd_cur,temp_dew_cur,tfmin=0.0;
    float rhsum=0.0,rhave,rhinit,twinit=0.0;
    int pw0,pw1,pw2,pw3,pw4,pw5,pw6,pw7,pw8,pw9,prewat,prewatflu;
    char glhrfls,dnrfls,dfhrfls,glhifls,dhifls,dnifls,rest[240];
    char zifls,tskycovfls,opqskycovfls,tdfls,twfls,rhfls,atmpfls,wdirfls;
    char wspdfls,visfls,ceilfls,prewatfls;

    read_inputs();
    del_x=0.02;  /* 2 cm spacing in nodes for 40 cm total */

    durcum=0;
    /* Compute cumulative days per month */
    days[1]=0;
    days[2]=31;
    days[3]=days[2]+29;
    days[4]=days[3]+31;
    days[5]=days[4]+30;
    days[6]=days[5]+31;
```

```
days[7]=days[6]+30;
days[8]=days[7]+31;
days[9]=days[8]+31;
days[10]=days[9]+30;
days[11]=days[10]+31;
days[12]=days[11]+30;

/* Get name of weather data file to use for this simulation */
printf("Enter name of file to read \n");
scanf("%s",filein);
printf("%s\n",filein);
infile=fopen(filein,"r");

/* Open log files for recording time-of-wetness and time-of-freezing */
wetfile=fopen("wettime.out","w");
frfile=fopen("freezetime.out","w");

/* Read in the header for this weather data file and echo values */
fscanf(infile,"%s %s %s %d %s %d %d %s %d %d %d",
  wba,city,state,&tzone,latd,&lat1,&lat2,longd,&long1,&long2,&elev);
printf("Number is %s\n",wba);
printf("City is %s\n",city);
printf("State is %s\n",state);
printf("Time zone is %d\n",tzone);

/* Set time step in finite difference routine */
/* based on thermal properties of concrete and soil */
del_t=del_x*del_x/4./alpha_conc;
del_t_soil=del_x*del_x/4./alpha_soil;
if(del_t_soil<del_t){del_t=del_t_soil;}
/* Scale time step to a factor of 3600 (seconds in an hour) */
if(del_t<50.0){
        del_t=10.;
}
else if((del_t>50.0)&&(del_t<100.0)){
        del_t=50.0;
}
else if((del_t>100.0)&&(del_t<200.0)){
        del_t=100.0;
}
else if((del_t>200.0)&&(del_t<300.0)){
        del_t=200.0;
}
else if((del_t>300.0)&&(del_t<400.0)){
        del_t=300.0;
}
else if(del_t>400.0){
```

```
                del_t=400.0;
        }
        printf("del_t is %lf \n",del_t);
        fflush(stdout);

        /* Read in the first record */
        /* Format statement is directly from TMY2DATA (NREL) Manual */
        fscanf(infile,"%2d%2d%2d%2d%4d%4d%4d%1s%1d%4d%1s%1d%4d%1s%1d%4d%1s%1d%4d
%1s%1d%4d%1s%1d%4d%1s%1d%2d%1s%1d%2d%1s%1d%4d%1s%1d%4d%1s%1d%3d%1s%1d%4d%1s%1d%3
d%1s%1d%3d%1s%1d%4d%1s%1d%5ld%1s%1d%1d%1d%1d%1d%1d%1d%1d%1d%1d%3d%1s%1d%s",
   &year,&month,&day,&hour,&exhr,&exdnr,&glhr,&glhrfls,&glhrflu,&dnr,&dnrfls,
   &dnrflu,&dfhr,&dfhrfls,&dfhrflu,&glhi,&glhifls,&glhiflu,&dni,&dnifls,&dniflu,
   &dhi,&dhifls,&dhiflu,&zi,&zifls,&ziflu,&tskycov,&tskycovfls,&tskycovflu,
   &opqskycov,&opqskycovfls,&opqskycovflu,&tempdry,&tdfls,&tdflu,&tempwet,
   &twfls,&twflu,&rh,&rhfls,&rhflu,&atmp,&atmpfls,&atmpflu,&wdir,&wdirfls,
   &wdirflu,&wspd,&wspdfls,&wspdflu,&vis,&visfls,&visflu,&ceil,&ceilfls,
   &ceilflu,&pw0,&pw1,&pw2,&pw3,&pw4,&pw5,&pw6,&pw7,&pw8,&pw9,&prewat,
   &prewatfls,&prewatflu,rest);
        /* Extract the relevant values */
        temp_amb_old=(double)tempdry/10.;
        wspd_old=(double)wspd/10.;
        temp_dew_old=(double)tempwet/10.;
        /* Initialize temperature data to ambient temperature value */
        for(j=0;j<=20;j++){
                tempold[j]=temp_amb_old;
        }
        tempold[20]=13.0;

        /* Now process each weather record in turn */
        for(i=1;i<8760;i++){
        fscanf(infile,"%2d%2d%2d%2d%4d%4d%4d%1s%1d%4d%1s%1d%4d%1s%1d%4d%1s%1d%4d
%1s%1d%4d%1s%1d%4d%1s%1d%2d%1s%1d%2d%1s%1d%4d%1s%1d%4d%1s%1d%3d%1s%1d%4d%1s%1d%3
d%1s%1d%3d%1s%1d%4d%1s%1d%5ld%1s%1d%1d%1d%1d%1d%1d%1d%1d%1d%1d%3d%1s%1d%s",
   &year,&month,&day,&hour,&exhr,&exdnr,&glhr,&glhrfls,&glhrflu,&dnr,&dnrfls,
   &dnrflu,&dfhr,&dfhrfls,&dfhrflu,&glhi,&glhifls,&glhiflu,&dni,&dnifls,
   &dniflu,&dhi,&dhifls,&dhiflu,&zi,&zifls,&ziflu,&tskycov,&tskycovfls,
   &tskycovflu,&opqskycov,&opqskycovfls,&opqskycovflu,&tempdry,&tdfls,&tdflu,
   &tempwet,&twfls,&twflu,&rh,&rhfls,&rhflu,&atmp,&atmpfls,&atmpflu,&wdir,
   &wdirfls,&wdirflu,&wspd,&wspdfls,&wspdflu,&vis,&visfls,&visflu,&ceil,
   &ceilfls,&ceilflu,&pw0,&pw1,&pw2,&pw3,&pw4,&pw5,&pw6,&pw7,&pw8,&pw9,&prewat,
   &prewatfls,&prewatflu,rest);
        /* Extract the relevant values */
        temp_amb=(double)tempdry/10.;
        temp_dew=(double)tempwet/10.;
        wspd_new=(double)wspd/10.;
        qsun=glhr;
        sum_t=0.0;
```

```
  /* Now process one hours worth of data */
  while(sum_t<3600.){
            /* Update the hourly and cumulative times */
            sum_t+=del_t;
            t_cum+=del_t;
            tfact=(sum_t)/3600.;

            /* Linearly interpolate the temperatures and wind speeds */
            /* within any 1 h period */
            temp_amb_cur=temp_amb_old+tfact*(temp_amb-temp_amb_old);
            temp_dew_cur=temp_dew_old+tfact*(temp_dew-temp_dew_old);
            wspd_cur=wspd_old+tfact*(wspd_new-wspd_old);
  /* Convection coefficient based on wind speed from FEMMASSE */
  /* of Intron in The Netherlands (3/00) */
            if(wspd_cur<5.0){
                  h_coeff=5.6+4.0*wspd_cur;
            }
            else{
                  h_coeff=7.2*pow(wspd_cur,0.78);
            }
 /* Call the routine to update all temperatures */
upd_temp(qsun,temp_amb_cur,temp_dew_cur,(double)tskycov/10.,del_x,del_t);
  }

  /* Now look for freezing and wetting events */
  temp_amb_old=temp_amb;
  temp_dew_old=temp_dew;
  wspd_old=wspd_new;
  /* Freeze events */
  if((freezeflag==0)&&(tempold[0]<(0.0))){
          freezeflag=1;
          timeforg=(day*24+days[month]*24+hour);
          tfmin=tempold[0];
  }
  /* Log minimum temperature achieved during freezing */
  if((freezeflag==1)&&(tfmin>tempold[0])){tfmin=tempold[0];}
  if((freezeflag==1)&&(tempold[0]>=(0.0))){
            freezeflag=0;
            timeffin=(day*24+days[month]*24+hour);
            duration=timeffin-timeforg;
            fprintf(frfile,"%ld %d %lf\n",timeforg,duration,tfmin);
   }

   /* Wet if rain (precipitation) or dew (condensation) event */
  if(((wetflag==0)&&(pw0==0)&&((pw2!=9)||(pw6!=9)||(pw3!=9)))||
      ((wetflag==0)&&((tempold[0]<temp_dew)&&(tempold[0]>0.0)))){
            wetflag=1;
```

```
               rhsum+=(float)rh;
         rhinit=(float)rh;
         twinit=tempold[0];
               printf("Wet from %d:00 %d/%d/%d with T= %lf  ",hour,
                     month,day,year,tempnew[0]);
               timeorg=(day*24+days[month]*24+hour);
        }
        if((wetflag==1)&&(pw0==0)&&((pw2==9)&&(pw6==9)&&(pw3==9))&&
             (tempold[0]>=temp_dew)){
               wetflag=0;
               printf("until %d:00 %d/%d/%d  ",hour,month,day,year);
               timefin=(day*24+days[month]*24+hour);
               duration=timefin-timeorg;
               durcum+=duration;
               numcum+=1;
               printf("of duration %d hours\n",duration);
               fprintf(wetfile,"%ld %d %f %f \n",timeorg,duration,
                      (twinit+tempold[0])/2.,rhinit);
        }
        }
        printf("\n");
     printf("total cumulative time of wetness is %ld in %d separate events\n",
         durcum,numcum);
        rhave=rhsum/(float)numcum;
        printf("average rh before start of rainfall is %f \n",rhave);
        fclose(infile);
        fclose(wetfile);
        fclose(frfile);
}
```