DEGDAY: A Program for Calculating Degree-days, and Assumptions Behind the Degree-day Approach

LEON G. HIGLEY,1 LARRY P. PEDIGO,1 AND KENNETH R. OSTLIE2


ABSTRACT A BASIC computer program (DEGDAY) for calculating degree-days on microcomputers is presented. The program calculates heating degree-days with the rectangle, triangle, and sine wave methods and calculates cooling degree days with the sine wave method. Assumptions and approximations associated with degree-day calculations are noted. In particular, eight factors affecting degree-days are discussed: 1) substrate availability, 2) enzyme availability, 3) approximations in laboratory estimates of development, 4) approximations in calculating the developmental minimum, 5) approximations in calculating a developmental maximum, 6) approximations in using single values for thresholds, 7) thermoregulation, and 8) limitations of temperature data. The various errors arising from these factors are discussed in relationship to error attributable to differences between actual and estimated degree-days.

KEY WORDS degree-day, developmental threshold, insect development, methodology, physiological time, thermal unit

THE PRINCIPLE OF using temperature and time to describe poikilotherm and plant development has been recognized for >250 years (Réaumur 1735), and modifications of the approach continue to be of interest. Degree-day models have become an established tool in plant sciences, pest management, and ecology and are useful in understanding insect and plant phenology, driving computer simulation models, and predicting pest status. Wang (1960) and Pruess (1983) reviewed the literature, described various calculation techniques, and critiqued the degree-day approach. Most recent interest in degree-days has focused on calculation techniques. Although these methods have become increasingly complex, the assumptions behind degree-day calculations have not changed. Unfortunately, by emphasizing calculation methods, other, potentially more significant, features of degree-day models are easily overlooked. Additionally, acceptance of newer calculation techniques has been hindered by the complexity of the calculations (Stark & Aliniazee 1982, Pruess 1983).

We have addressed these problems by reviewing the assumptions and approximations associated with degree-day calculations. Moreover, we have developed a program for microcomputers, DEGDAY, that calculates degree-days by using three different techniques.

Assumptions in Degree-day Modeling

The essential assumption implicit in all degree-day models is that plant and poikilotherm development is directly related to ambient temperature and time. Certainly, all development depends on time; growth is not instantaneous. However, the relationship of temperature to growth is not as straightforward. At a basic level, development in an organism depends on the rates of various biochemical reactions. As these reactions occur through time, development proceeds.

Degree-day models use the assumption that one, or a number, of these enzyme-catalyzed reactions are "rate limiting" for growth (Barnes 1937). The individual rates for these reactions depend on three criteria: availability of substrates, availability of enzymes, and temperature. Development is retarded when substrates—water, nutrients, and photosynthates for plants, water and food for animals—are not readily available (Lees 1955, Higley & Pedigo 1984b). Therefore, all required substrates (i.e., nutrients) must be present in adequate quantities for optimal growth.

Reduced availability of enzymes also leads to reduced growth rates. Because enzyme concentrations may be regulated hormonally (through factors such as photoperiod or genetics), growth in some plants and poikilothermic organisms does not follow degree-day accumulations well. Additionally, fluctuating temperatures can change which reactions are favored, with consequent changes and lags in the enzyme and substrate availability. Thus, development under a changing temperature regime often is different than under constant temperatures (Howe 1967). The magnitude of the differences between development times under constant and fluctuating temperatures depends on the average temperature, amplitude, and frequency of the fluctuations (Campbell et al. 1974).

Availability of substrates and enzymes is impor-
tant when considering how species-specific development times are determined. For example, development times (in degree-days) for insects are usually calculated in growth chambers with optimal diet and constant temperatures. Consequently, these development times must be regarded as minimums. Moreover, because factors such as photoperiod often are not examined, temperature improperly may be assumed to be the primary determinant of growth rate, whereas other factors may be critical under field conditions.

Within limits, higher temperatures produce greater growth rates because reactions proceed more rapidly at higher temperatures. As temperature increases, diffusion rates for substrates or enzymes or both also increase, resulting in greater formation of enzyme/substrate complexes. Additionally, higher temperatures provide more thermal energy for meeting energy requirements of the reactions (Barnes 1937, Metzler 1977). The relationship between temperature and growth rate is usually calculated as linear, but it is actually curvilinear (Fig. 1). Sharpe & DeMichele (1977) presented a thermodynamic model of poikilotherm development that describes the biochemical basis for the curve. They maintained that nonlinear portions of the curve are a consequence of inactivation of a control enzyme, or enzymes, at high and low temperatures (by producing conformational changes in the enzyme[s]), with a linear response at intermediate temperatures. Algorithms have been developed using a variety of nonlinear functions that characterize the temperature/growth rate relationship (Stinner et al. 1974, Logan et al. 1976, Wagner et al. 1984), but for most species, the linear approximation is acceptable (as long as the temperatures considered are in the linear region). However, the kind of approximation becomes important when we consider the extremes of the temperature/growth rate relationship.

Specifically, for a given species a developmental minimum temperature is determined; this is the lowest temperature at which the rate-limiting reaction or reactions will occur. This temperature is usually calculated by measuring development over a range of temperatures, fitting these growth rates to a line, and then extrapolating to zero. This technique, known as the x-intercept method (Arnold 1959), obviously produces only an estimate of the developmental minimum. Other approaches are available for calculating the minimum threshold (Arnold 1959, Kirk & Alinazee 1981), but they also produce estimated values. Thus, in calculating degree-days, the minimum threshold invariably introduces some inaccuracy.

An analogous threshold occurs at the upper end of the scale, the developmental maximum temperature. This is the temperature at which the development rate reaches its maximum (where kinetic constraints limit enzyme and substrate diffusion and beyond which temperatures are sufficiently high as to produce structural changes in control enzymes, to impair general enzyme function, or to actually denature some proteins (Metzler 1977, Sharpe & DeMichele 1977); not surprisingly, the developmental maximum often approaches the lethal temperature for a species). Estimating a developmental maximum is challenging because the variability in developmental rates is usually greater at higher temperatures and because mortality is high. Techniques for calculating developmental maxima are not precise; however, one rule of thumb often used is to choose the developmental maximum as the upper temperature beyond which the growth rate plateaus or declines. This point may be estimated (Funderburk et al. 1984) or calculated iteratively (Nowierski et al. 1983).

Most often, developmental maxima are not determined. Without a developmental maximum, no upper bound is placed on daily temperatures used to calculate degree-days and the accumulations may indicate greater development than actually occurred. Although the lack of a developmental maximum may make degree-day calculations less accurate, the error introduced is not too great if the daily maximum temperatures are usually below the developmental maximum. For example, 1982 accumulations for Fairbanks, Alaska (Table 1), without a developmental maximum were only slightly higher than accumulations with a maximum. However, substantial differences can occur when daily maximum temperatures frequently exceed the developmental maximum. Thus, for Phoenix, Ariz., and Orlando, Fla., the 1982 accumulations without a developmental maximum are much greater (10-20%) than accumulations calculated with a developmental maximum. These examples illustrate that the error in degree-day calculations introduced by not including a developmental maximum depends on how much and how frequently the daily maximum temperatures
are above the developmental maximum. Moreover, it follows that the lower the developmental maximum, the greater is the error introduced by not including the maximum in the calculations (e.g., see accumulations with 25 versus 30°C developmental maximum, Table 1).

The developmental maximum and minimum temperatures are calculated as single values, but the actual minima and maxima will vary with the age of the organism (Wang 1960). Insects, for example, may have different threshold temperatures for each of the different life stages (e.g., egg, larva, pupa, and adult) (Sanborn et al. 1982). Despite these differences in threshold with age, single values (averages) for thresholds are most often used (to avoid undue complexity).

Two additional assumptions implicit in using degree-days to model development are that the organism cannot regulate its own temperature (i.e., body temperature follows ambient temperature), and the temperature data used for calculating degree-days represent the same temperatures as the organism experienced. Many poikilothermic organisms violate this first assumption. Among insects, for example, species use behavioral and physiological mechanisms for thermoregulation (May 1979). One common behavioral mechanism is for an organism to seek a thermally favored microhabitat. Such behavior clearly relates to the second assumption, that temperature data for calculations reflect actual temperatures experienced by the organism in question. Obviously, differences between microhabitat and ambient air temperatures produce inaccuracies in degree-day estimates. Similarly, temperature data from a single site (such as a weather service recording station) can only approximate temperatures at other locations.

One further problem in using degree-days is determining when to begin the accumulations. A common approach is to begin accumulations once temperatures exceed the developmental minimum, but development may not occur until later. Frequently, initiating accumulations requires estimating when diapause has terminated; Tauber et al. (1986) summarize approaches to this problem. For migratory species, it may be possible to initiate accumulations to correspond to a given phenological event (e.g., capture of parous females in light traps).

As this discussion illustrates, degree-day accumulations for a species are influenced by eight factors: 1) substrate availability (nutritional deficiencies, effects of fluctuating temperatures); 2) enzyme availability (hormonal effects on growth, effects of fluctuating temperatures); 3) approximations and assumptions in laboratory estimates of development; 4) approximations in calculating the developmental minimum; 5) approximations in calculating (or not calculating) the developmental maximum; 6) approximations in using single values for developmental thresholds (rather than changing thresholds with age); 7) thermoregulation; and 8) propriety and limitations of temperature data used in calculations.

Although the use of degree-days requires making many assumptions and approximations, the approach is sound, as long as temperature is the major determinant of growth rate and the limitations of the approach are recognized. In particular, the

---

Table 1. Celsius degree-day calculation comparisons for January through 31 December 1982 at five National Oceanic and Atmospheric Administration weather stations: Fairbanks WSFO AP, Alaska; Phoenix WSFO AP, Ariz.; Orlando WSO MC Coy, Fla.; Ames WSW, Iowa; and Lansing WSO AP, Mich. (5°C developmental minimum; triangle and sine wave methods calculated by half days unless otherwise indicated)

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Rectangle</td>
<td>1,239</td>
<td>6,598</td>
<td>6,656</td>
<td>2,773</td>
<td>523</td>
</tr>
<tr>
<td>None</td>
<td>Triangle</td>
<td>1,210</td>
<td>6,579</td>
<td>6,656</td>
<td>2,684</td>
<td>293</td>
</tr>
<tr>
<td>None</td>
<td>Sine wave</td>
<td>1,221</td>
<td>6,583</td>
<td>6,640</td>
<td>2,707</td>
<td>202</td>
</tr>
</tbody>
</table>

(Deviation from accumulations without developmental maximum in parentheses)

25° Rectangle 1,238 (-21) 5,275 (-1,323) 5,843 (-813) 2,026 (-147) 2,440 (-279)
25° Triangle 1,204 (-6) 5,459 (-1,190) 6,124 (-512) 2,028 (-42) 2,443 (-190)
25° Sine wave 1,211 (-10) 5,412 (-1,171) 6,002 (-588) 2,027 (26) 2,444 (-211)
30° Rectangle 1,259 (0) 5,903 (-695) 6,455 (-201) 2,759 (-4) 2,552 (-67)
30° Triangle 1,210 (0) 6,092 (-487) 6,579 (-37) 2,585 (-5) 2,485 (-148)
30° Sine wave 1,221 (0) 6,044 (-539) 6,547 (-93) 2,521 (-6) 2,503 (-152)

Whole-day calculations (deviation from half-day calculations in parentheses)

None Triangle 1,206 (-4) 6,581 (2) 6,642 (6) 2,688 (4) 2,634 (1)
None Sine wave 1,217 (-4) 6,585 (2) 6,645 (5) 2,690 (2) 2,655 (1)
25° Triangle 1,201 (-3) 5,463 (4) 6,199 (5) 2,631 (3) 2,444 (1)
25° Sine wave 1,208 (-5) 5,415 (3) 6,007 (5) 2,639 (3) 2,445 (1)
30° Triangle 1,206 (-4) 6,095 (3) 6,585 (6) 2,685 (6) 2,487 (2)
30° Sine wave 1,217 (-4) 6,046 (2) 6,558 (3) 2,703 (3) 2,504 (1)
biological rationale and practical considerations behind degree-day calculations are such that degree-days must always be interpreted as estimates of developmental time.

One method advocated to avoid some of these problems is to use curvilinear functions to describe development rate rather than degree-days, which is essentially a linear approach. Wagner et al. (1984) present an excellent summary of various functions used to describe insect development rates. These authors concluded that Sharpe & DeMichele's (1977) model offers the best results for predicting development times, and they provided a computer program for using this model. However, although curvilinear functions address problems with developmental maxima and minima, they are still subject to the other limitations previously mentioned. Moreover, curvilinear models do not possess any a priori advantage over degree-day methods. For instance, Hochberg et al. (1986) found that a degree-day model offered better estimates of pea aphid, *Acyrthosiphon pisum*, and blue alfalfa aphid, *A. kondoi*, development than more involved models, which "challenges the commonly held view that the more parameters a model has the better its performance will be." When temperatures are in the nonlinear portion of the development curve, curvilinear models are clearly better predictors, but for many, possibly most, uses, degree-day methods are as accurate (and easier to calculate).

In calculating degree-days, we cannot avoid the central requirement that degree-days are the major determinant of growth. But in certain applications, some approximations can be avoided. For example, when degree-days are used in computer simulation models, many factors affecting degree-days may be addressed explicitly: different (age-specific) developmental thresholds can be used; the influence of nutrition and fluctuating temperatures can be included. Such complexities are easily incorporated into computer models if experimental data on these factors are available. But this level of sophistication is un warranted if the data used for calculating degree-days are not comparably precise.

Indeed, we believe that temperature data are the chief limitation to our producing more accurate degree-day accumulations. Whereas most other approximations in degree-day calculations can be described through experimentation and included in calculations (though adding considerable complexity), we cannot improve our temperature estimates by short of monitoring internal body temperatures of many free-living individuals in their natural environment. Baker et al. (1985) discuss many of the discrepancies in temperature data between weather stations including different observation times, latitudes, surfaces, topography, and urbanization. This discussion emphasizes that temperature data from one, or even several, sites can provide only coarse estimates of the thermal environment for organisms spread over many square kilometers. Additionally, thermoregulation and differences between ambient and microhabitat temperatures further complicate the problem.

Clearly, various errors are introduced into degree-day estimates. Frequently, the magnitudes of these errors cannot be determined easily; in fact, errors may cancel out. When a degree-day model shows a consistent bias, however, it may be possible to empirically adjust the model. Indeed, degree-day models can be developed entirely from field data (e.g., Ring & Harris 1983, Hochberg et al. 1986). Nevertheless, the ultimate resolution of any degree-day model is limited by variability in development rates within a population (although we would anticipate a normal distribution around theoretical accumulations). The necessary resolution, or accuracy, of any degree-day model depends on the intended use of the model. In making predictions in pest management, for example, estimates within 10–15% may be adequate. Alternatively, in using degree-days to study population dynamics, more precise estimates are desirable. However, in determining population dynamics and phenologies, the need for precise degree-day estimates is less important than the need to use degree-days in conjunction with direct age-grading methods, such as ovarian dynamics (Tyndale-Biscoe 1984).

### Degree-day Calculation Techniques

Given this background, we can address the question of degree-day calculation techniques. Determining the degree-day accumulation for a date involves calculating the area above the developmental threshold and under the temperature curve for that date. A number of algorithms have been devised that use only daily maximum and minimum temperatures to estimate this area; all make assumptions about the shape of the temperature curve. It might be argued that we should strive for exact degree-day estimates to avoid introducing additional error into an already inaccurate index. Certainly, recent interest in degree-day calculations has focused on developing calculation methods that more accurately estimate the area under the temperature curve. However, even if we monitored temperatures continually and exactly measured the area under the curve, we would not produce "perfect" degree-days because, implicitly, degree-days are estimates. In fact, as Pruess (1983) noted, such "actual" degree-days, representing the exact area under the temperature curve, have no more biological validity than their estimated counterparts.

This question of calculated versus actual degree-days, and the related question of which calculation method to use, directly relate to our consideration of errors. A law of error states that "small errors
do not matter until large errors are removed" (Landes 1983). If we consider the errors previously discussed, such as limitations in temperature data or approximations in calculating developmental maxima and minima, then the discrepancy between calculated and actual degree-days may seem trivial by comparison. Unfortunately, these various errors are difficult to quantify, so we are not certain that these are "large errors," and, consequently, we cannot neglect differences between calculation methods.

In the most commonly used method, area under the curve is calculated as a rectangle (Arnold 1959). The original rectangle technique did not provide for a developmental maximum, but Baskerville & Emin (1969) modified the approach to accommodate a maximum. The algorithm for the rectangle method is presented in lines 1570–1640 of the DEGDAY program (Appendix 1).

Area under the curve also has been estimated as a triangle (Lindsey & Newman 1956). This approach was modified by Sevacherian et al. (1977) to recognize that the second daily minimum may not be the same as the first. Thus, the triangle method calculates area under the curve on a half-day basis. (For half days, the area measured is usually that of a trapezoid, rather than a triangle.) Stark & Alinazee (1982) presented a lucid explanation of the triangle method and adapted it to include a developmental maximum.

A third approach is to approximate the temperature curve as a sine wave and then calculate area under this sine wave curve. Arnold (1960) provided an early description of this method, and Allen (1976) modified the technique and presented formulas and a computer program for its use. Allen also presented a technique called bias correction, which accounts for differences in geographic location (latitude) that can affect sine wave estimates.

Formulas for the triangle and sine wave methods are presented in lines 2060–2140 of the DEGDAY program. These formulas vary, depending on the specific relationship between the daily maximum and minimum and the developmental maximum and minimum. In particular, the six possible events are as follows (Allen 1976):

1) daily minimum < developmental maximum;
2) daily maximum ≤ developmental maximum;
3) daily minimum ≥ developmental minimum and daily maximum ≤ developmental maximum;
4) daily minimum < developmental minimum and developmental minimum < daily maximum ≤ developmental maximum;
5) daily maximum > developmental maximum and developmental maximum > daily minimum ≤ developmental minimum; and
6) daily minimum < developmental minimum and daily maximum > developmental maximum.

Only events 2–4 can occur if calculations are made without a developmental maximum. Additionally, degree-days calculated for events 1–3 will be the same, regardless of which of the three calculation methods is used. In terms of degree-days calculated by the three methods: for event 4, rectangle > sine wave > triangle; for event 5, triangle > sine wave > rectangle; and for event 6, rectangle > sine wave > triangle. Triangle and sine wave degree-days are generally closer for events 4–6 than are sine wave and rectangle (rectangle and triangle are the most different).

Both the triangle and sine wave methods calculate degree-days by half days (the first 12-h accumulation is calculated with that day's maximum and minimum, and the second 12-h accumulation is calculated with that day's maximum and the next day's minimum). Pruess (1983) argued that calculation by half day implies a gain in precision which does not exist. We examined differences between half- and whole-day calculation techniques. (For these comparisons we used 1982 data from locations comparable to those used by Allen [1976] and Pruess [1983]) Table 1 indicates that deviations between the half- and whole-day calculations are relatively trivial (ca. 1–3% difference over 1 year). (Positive and negative differences between calculations by whole- and half-day tend to cancel through time.) Consequently, on a practical level, we agree with Pruess and do not see any need to choose between the two approaches; however, because the triangle and sine wave methods were originally described as half-day techniques, we believe that calculations for these methods should be made by half day for consistency. DEGDAY uses half-day calculations for triangle and sine wave methods, but the alternative code for whole-day calculations is presented (Appendix 3).

Pruess (1983) also suggested that in the absence of actual degree-days sine wave estimates should be the method of choice. He noted that sine wave estimates are closer to actual degree-days when minimum daily temperatures are below the developmental minimum and are, therefore, most appropriate for describing development in the spring. Sine wave estimates provided better predictions of seedcorn maggot, Delta platura, vernal development than rectangle estimates (Higley & Pedigo 1984a), but rectangle estimates were superior to sine wave in describing vernal development of pecan nut casebearer, Acrobasis nuxvorella (Ring & Harris, 1983).

DEGDAY Program

In developing the DEGDAY program, we decided not to choose one calculation method over another; DEGDAY calculates heating degree-days by the rectangle, triangle, and sine wave methods and calculates cooling degree-days by the sine wave method. (Cooling degree-days represent accumu-
3030, 4030, and 5030 require this change, as do grams. In DEGDAY, lines 530, 1030, 1330, 2530, by whole day; it can be used to replace lines 1660,..,
calculation of the triangle and sine wave methods (which was our motivation in developing DEGDAY). Although the length of the DEGDAY program may be imposing, it is a consequence of including full annotation and making the program easy to use.

Appendix 1 contains an annotated listing of the DEGDAY program, and Appendix 2 contains a listing of a program with instructions for using DEGDAY. The file names used for these programs should be DEGDAY.BAS and DDINSTR.BAS, respectively, to permit chaining between the two programs. On systems with >64K memory, these two programs can be combined with slight modification (line 7000 in DEGDAY should be deleted, and line 8570 in DDINSTR should be changed to a RETURN). Appendix 3 contains code for calculations of the triangle and sine wave methods by whole day; it can be used to replace lines 1660–2140 in DEGDAY.

DEGDAY was developed on a KAYPRO 2 microcomputer (280A 4 MHz cpu, 64K RAM). Execution time for calculating degree-days for 1 year (366 days input data) is ca. 4 min. The programs presented are written in Microsoft BASIC for microcomputers using the CP/M operating system. The only modification required for running DEGDAY with versions of Microsoft BASIC for the MS-DOS (PC-DOS) operating system is to replace “PRINT CHR$(26)” with “CLS” throughout both programs. In DEGDAY, lines 530, 1030, 1330, 2530, 3030, 4030, and 5030 require this change, as do lines 7000, 7210, 7360, 7580, 7820, 8060, 8170, and 8540 in DDINSTR. PRINT CHR$(26) and CLS are the commands to clear the screen for CP/M and MS-DOS versions of Microsoft BASIC, respectively.

In the entering the program, comment lines (lines starting with an apostrophe) can be eliminated, but line numbering should not be altered. For proper output, all “LPRINT USING” statements should be entered as shown. A complete description of DEGDAY’s operation, instructions on using the program, and data set requirements are provided in Appendix 2.

Acknowledgment

We thank W. C. Bailey (Dep. of Entomology, Univ. of Missouri, Columbia) for testing initial versions of DEGDAY and suggesting modifications. This is Journal Paper No. J-12049 of the Iowa Agric. and Home Econ. Exp. Stn., Ames; Project No. 2580.

References Cited


References Cited


Received for publication 21 October 1985; accepted 23 June 1986.
Appendix 1. DEGDAY Program

10 ' ***** DEGDAY - DEGREE DAY-CALCULATION PROGRAM *****
20 ' Ver 2.8 - May 1986
30 ' ***** Variable List *****
40 ' ACCUM(I)= array for accumulations between dates
50 ' C= index variable; 1=Celsius 2=Fahrenheit
60 ' COOL(I,J)= array for daily cooling degree-days
70 ' D(I,J)= input data array
80 ' FILENAME$= input data filename
90 ' FNCTOF(X)= function to convert Celsius to Fahrenheit
100 ' FILENAMES= results filename
110 ' HCOOL(I,J)= half-day cooling degree-days
120 ' HSINE(I,J)= half-day sine wave degree-day accumulation
130 ' HTRI(I,J)= half-day triangle degree-day accumulation
140 ' IFIRSTDATE= first date in input data (temperature) file
150 ' ILASTDATE= last date in input data (temperature) file
160 ' LINECOUNT= index variable for output line number (to be printed next)
170 ' MENU= menu index variable
180 ' MT= mean daily temperature
190 ' PROCEED$= index variable for testing whether to continue
200 ' R(I,J)= results array
210 ' RFILENAME$= results filename
220 ' STARTOATE= date to start degree-day accumulation
230 ' TOPDATE= date to stop degree-day accumulation
240 ' TITLE$= title for printed output
250 ' TLO= lower developmental threshold
260 ' TMAX= daily maximum temperature
270 ' TMIN= daily minimum temperature
280 ' TRI(I,J)= daily triangle degree-day accumulation
290 ' TUP= upper developmental threshold
300 ' TTYPE$= temperature scale for TUP and TL0
310 ' WRECT(I,J)= daily rectangle degree-day accumulation
320 ' A, T1, T2, X1, X2 = factors for sine wave calculations
330 ' E, I, IN, J, K, L = index variables
340 ' ***** Program Menu *****
350 ' OPTION BASE 1
360 ' DEFINT D,R
370 ' DIM D(367,3),R(367,17)
380 ' DEF FNCTOF(X)=(9/5)*X+32
390 ' DEF FNFTOC(X)=(5/9)*(X-32)
400 ' PRINT CHR$(26)
410 ' PRINT " DEGDAY - Degree-day Calculation Program"
420 ' PRINT " 1. Calculate Degree-days (heating and cooling)"
430 ' PRINT " 2. Access Results File"
440 ' PRINT " 3. Print Results File"
Appendix 1. Continued

4. Calculate Degree-days Accumulated Between Dates
5. Instructions
6. Exit to Operating System

Appendix Input Prompts

605 PRINT "Enter input data filename: ", FILENAMES: PRINT: PRINT
607 OPEN "I", l, FILENAMES
608 INPUT "Enter number from menu: ", MENU
609 ON ERROR GOTO 0
610 FOR 1 = 1 TO 367
611 INPUT D(I,1), D(I,2), D(I,3)
612 ILASTDATE = D(I,1)
613 IF EOF(l) THEN I = 367
614 INPUT "Enter the initial Julian date for accumulation: ", STARTDATE
615 INPUT "Enter the final Julian date for accumulation: ", STOPDATE
616 IF (IFIRSTDATE) STARTDATE) OR (ILASTDATE <= STOPDATE) THEN PRINT "The starting or stopping date you entered is outside the range of dates in "; FILENAMES "; Try again."
617 IF STOPDATE <= STARTDATE THEN PRINT: PRINT "The stopdate you entered is less than the startdate. Try again."
618 IF DTYPES <> "C" AND DTYPES <> "F" THEN PRINT "Incorrect entry. Use only C or F."
619 INPUT "Enter developmental minimum: ", TLO
620 INPUT "Do you want to use a developmental maximum (Y or N)? "; PROCEEDS
621 IF PROCEEDS = "Y" THEN INPUT "Enter developmental maximum: ", TUP ELSE TUP = 1000
622 IF TUP <= TLO THEN PRINT "The developmental maximum is <= to the developmental minimum which is incorrect.". PRINT Please check your developmental thresholds and try again.
623 PRINT "Input values are:": PRINT "Input data in degrees "; DTYPES; " from file "; FILENAMES
624 IF PROCEEDS = "Y" THEN PRINT "Developmental maximum of "; TUP; " degrees "; TTYTYPE; ELSE PRINT "No developmental maximum."
625 PRINT "Accumulation calculated from Julian date "; STARTDATE; " to "; STOPDATE
626 IF these values are not correct enter 'Y', otherwise hit <RETURN> ", PROCEEDS
Appendix 1. Continued

1390 IF PROCEED$="N" THEN GOTO 1000
1400 PRINT : PRINT "Calculations in progress"
1410 '**** test for temp scales and convert if needed
1420 '1430 '1440 IF DTYPES="C" THEN C=1 ELSE C=2
1450 IF DTYPES=TTYPES GOTO 1530
1460 IF TTYPES="C" THEN TLO=FNCTOF(TLO) : TUP=FNCTOF(TUP) ELSE TLO=FNTOC(TLO) : TUP=FNTOC(TUP)
1470 '1480 '1490 '1500 IF R(1,1)<0 THEN ERASE R : DIM R(366,17)
1510 DIM COOL(2,2),HCOOL(2),HSINE(2),HTRI(2),SINE(2),TRI(2,2),WRECT(2,2)
1520 '1530 FOR J=1 TO (STOPDATE-STARTDATE)+1 'Loop for calculating DD
1540 IN=(STARTDATE-IFIRSTDATE)+I
1550 TMAX=D(IN,2) : TMIN=D(IN,3)
1560 '1570 '1580 '1590 IF TMIN<TLO THEN TMIN=TLO
1600 IF TMIN>TUP THEN TMIN=TUP
1610 IF TMAX<TLO THEN WRECT(C,1)=0 GOTO 1640
1620 IF TMAX>TUP THEN TMAX=TUP
1630 WRECT(C,1)=(TMAX+TMIN)/2-TLO
1640 WRECT(C,2)=WRECT(C,2)+WRECT(C,1)
1650 '1660 '1670 '1680 FOR J=1 TO 2
1690 TMAX=D(IN,2) : IF J=1 THEN TMIN=D(IN,3) ELSE TMIN=D(IN+1,3)
1700 '**** test for cases 1-6
1710 '1720 IF TMIN<TLO AND TMAX<TLO : TUP=TMIN GOTO 1780
1730 IF TMIN<TLO AND TMAX<TUP : TUP=TMIN GOTO 1790
1740 IF TMIN<TUP AND TMAX<TMIN GOTO 1800
1750 IF TMIN<TUP AND TMAX<TUP GOTO 1810
1760 IF TMIN>TLO AND TMAX>TLO GOTO 1820
1770 IF TMIN>TLO AND TMAX>TUP GOTO 1830
1780 '**** Case 1
1790 HTRI(J)=(TUP-TLO)/2
1800 HSINE(J)=(TUP-TLO)/2
1810 HCOOL(J)=0
1820 GOTO 2140
1830 '**** Case 2
1840 HTRI(J)=0
1850 HSINE(J)=0
1860 HCOOL(J)=(TLO-A)/2
1870 GOTO 2140
1880 '**** Case 3
1890 HTRI(J)=(MT-TLO)/2
1900 HSINE(J)=(MT-TLO)/2
1910 HCOOL(J)=0
1920 GOTO 2140
1930 '**** Case 4
Appendix 1. Continued

```
1940 HTRI(J)=(TMAX-TLO)^2/((TMAX-TMIN)*4)
1950 X1=(TLO-MT)/A
1960 TI=ATN(X1/SQR(1-X1^2))
1970 T2=1.5708
1980 GOTO 2110 '*** goto accumulation formula ***
1990 '**** Case 5 ****
2000 HTRI(J)=(MT-TLO)/2-(TMAX-TUP)^2/((TMAX-TMIN)*4)
2010 X2=(TUP-MT)/A
2020 T1=-1.5708
2030 T2=ATN(X2/SQR(1-X2^2))
2040 GOTO 2110 '*** goto accumulation formula ***
2050 '**** Case 6 ****
2060 HTRI(J)=(TMAX-TLO)^2-(TMAX-TUP)^2)/((TMAX-TMIN)*4)
2070 X1=(TLO-MT)/A
2080 X2=(TUP-MT)/A
2090 TI=ATN(X1/SQR(1-X1^2))
2100 T2=ATN(X2/SQR(1-X2^2))
2110 '**** Calculation formula for cases 4-6 ****
2120 HSINE(J)=159155*((TLO-MT)+A*(COS(T1)-COS(T2))+TUP-TLO)*(1.5708-T2)
2130 HCOOL(J)=159155*((TLO-MT)*(T1+1.5708)+A*COS(T1))
2140 NEXT J
2150 '**** Total half-day accumulations ****
2160 TRI(C,1)=HTRI(1)+HTRI(2)
2170 TRI(C,2)=TRI(C,1)+TRI(C,1)
2180 SINE(C,1)=HSINE(1)+HSINE(2)
2190 SINE(C,2)=SINE(C,2)+SINE(C,1)
2200 COOL(C,1)=HCOOL(1)+HCOOL(2)
2210 COOL(C,2)=COOL(C,2)+COOL(C,1)
2220 IF C=1 THEN E=2 : G=9/5 ELSE E=1 : G=5/9
2230 FOR J=1 TO 2
2240 WRECT(E,J)=G*WRECT(C,J)
2250 TRI(E,J)=G*TRI(C,J)
2260 SINE(E,J)=G*SINE(C,J)
2270 COOL(E,J)=G*COOL(C,J)
2280 NEXT J
2290 R(I,1)=D(IN,1) : L=1
2300 FOR J=1 TO 2
2310 FOR K=1 TO 2
2320 L=L+1
2330 R(I,L)=WRECT(J,K)
2340 R(I,L+4)=TRI(J,K)
2350 R(I,L+8)=SINE(J,K)
2360 R(I,L+12)=COOL(J,K)
2370 NEXT K
2380 NEXT J
2390 NEXT I
2400 ERASE COOL,HCOOL,HSINE,HTRI,SINE,TRI,WRECT
2410 RFIRSTDATE=STARTDATE : RLASTDATE=STOPDATE
2420 PRINT : PRINT "Calculations completed"
2430 PRINT CHR$(13)+CHR$(27)
2440 INPUT "Do you want a printed copy of the results (Y or <N>):" ;PROCEED$ 
2450 IF PROCEED$="Y" THEN GOSUB 5000
2460 INPUT "Do you want to store the results on disk (Y or <N>):" ;PROCEED$ 
2470 IF PROCEED$="Y" THEN GOSUB 2550
2480 INPUT "Do you want to calculate accumulations between dates from results (Y or <N>):" ;PROCEED$ 
2490 IF PROCEED$="Y" THEN GOSUB 2550
2500 IF PROCEED$="N" THEN GOSUB 5000
```

---

Appendix 1. Continued

```
1940 HTRI(J)=(TMAX-TLO)^2/((TMAX-TMIN)*4)
1950 X1=(TLO-MT)/A
1960 TI=ATN(X1/SQR(1-X1^2))
1970 T2=1.5708
1980 GOTO 2110 '*** goto accumulation formula ***
1990 '**** Case 5 ****
2000 HTRI(J)=(MT-TLO)/2-(TMAX-TUP)^2/((TMAX-TMIN)*4)
2010 X2=(TUP-MT)/A
2020 T1=-1.5708
2030 T2=ATN(X2/SQR(1-X2^2))
2040 GOTO 2110 '*** goto accumulation formula ***
2050 '**** Case 6 ****
2060 HTRI(J)=(TMAX-TLO)^2-(TMAX-TUP)^2)/((TMAX-TMIN)*4)
2070 X1=(TLO-MT)/A
2080 X2=(TUP-MT)/A
2090 TI=ATN(X1/SQR(1-X1^2))
2100 T2=ATN(X2/SQR(1-X2^2))
2110 '**** Calculation formula for cases 4-6 ****
2120 HSINE(J)=159155*((TLO-MT)+A*(COS(T1)-COS(T2))+TUP-TLO)*(1.5708-T2)
2130 HCOOL(J)=159155*((TLO-MT)*(T1+1.5708)+A*COS(T1))
2140 NEXT J
2150 '**** Total half-day accumulations ****
2160 TRI(C,1)=HTRI(1)+HTRI(2)
2170 TRI(C,2)=TRI(C,1)+TRI(C,1)
2180 SINE(C,1)=HSINE(1)+HSINE(2)
2190 SINE(C,2)=SINE(C,2)+SINE(C,1)
2200 COOL(C,1)=HCOOL(1)+HCOOL(2)
2210 COOL(C,2)=COOL(C,2)+COOL(C,1)
2220 IF C=1 THEN E=2 : G=9/5 ELSE E=1 : G=5/9
2230 FOR J=1 TO 2
2240 WRECT(E,J)=G*WRECT(C,J)
2250 TRI(E,J)=G*TRI(C,J)
2260 SINE(E,J)=G*SINE(C,J)
2270 COOL(E,J)=G*COOL(C,J)
2280 NEXT J
2290 R(I,1)=D(IN,1) : L=1
2300 FOR J=1 TO 2
2310 FOR K=1 TO 2
2320 L=L+1
2330 R(I,L)=WRECT(J,K)
2340 R(I,L+4)=TRI(J,K)
2350 R(I,L+8)=SINE(J,K)
2360 R(I,L+12)=COOL(J,K)
2370 NEXT K
2380 NEXT J
2390 NEXT I
2400 ERASE COOL,HCOOL,HSINE,HTRI,SINE,TRI,WRECT
2410 RFIRSTDATE=STARTDATE : RLASTDATE=STOPDATE
2420 PRINT : PRINT "Calculations completed"
2430 PRINT CHR$(13)+CHR$(27)
2440 INPUT "Do you want a printed copy of the results (Y or <N>):" ;PROCEED$ 
2450 IF PROCEED$="Y" THEN GOSUB 5000
2460 INPUT "Do you want to store the results on disk (Y or <N>):" ;PROCEED$ 
2470 IF PROCEED$="Y" THEN GOSUB 2550
2480 INPUT "Do you want to calculate accumulations between dates from results (Y or <N>):" ;PROCEED$ 
2490 IF PROCEED$="Y" THEN GOSUB 2550
2500 IF PROCEED$="N" THEN GOSUB 5000
```
Appendix 1. Continued

2490 IF PROCEED$="Y" THEN GOSUB 3000
2500 PRINT "Do you want to calculate additional accumulations with data from ":FILENAMES$" ?
2510 PRINT "(You can change developmental thresholds, start and stop dates.)"
2520 INPUT "Y or <N>: ",PROCEED$
2530 IF PROCEED$="Y" THEN PRINT CHR$(26) : GOTO 1210
2540 RETURN
2550 ' ****** Store Results File ******
2560 INPUT "Enter filename for results file: ",RFILENAME$
2570 ON ERROR GOTO 6000
2580 OPEN "O",l,RFILENAME$
2590 ON ERROR GOTO 0
2600 WRITE#l,TUP,TLO,C
2610 FOR I=1 TO (RLASTDATE-RFIRSTDATE)+1
2620 FOR J=1 TO 17
2630 WRITE#l,R(I,J)
2640 NEXT J
2650 NEXT I
2660 CLOSE#1
2670 PRINT "FILENAMES$ stored on disk." PRINT
2680 RETURN
2590 ' **** Calculate Accumulations Between Dates ****
2600 DIM ACCUM(8) : LINECOUNT=0
2610 PRINT "Enter title for output (80 char maximum, do not use commas)."
2620 INPUT TITLE$
2630 N=40-LEN(TITLE$)/2
2640 WHILE PROCEED$O"N"
2650 INPUT "Enter initial date: ",STARTDATE
2660 INPUT "Enter final date: ",STOPDATE
2670 IF STOPDATE<=STARTDATE THEN PRINT: PRINT "The final date must be > the initial date. Please try again." : PRINT: GOTO 3110
2680 IF STARTDATE<RFIRSTDATE OR STOPDATE>(RLASTDATE-l) THEN PRINT "The initial or final date is beyond the dates in the results. Please try again." : GOTO 3110
2690 IF STARTDATE=RFIRSTDATE THEN ACCUM(I)=R(SOPEDATE-RFIRSTDATE)+1,1+2*I) ELSE ACCUM(I)=R(SOPEDATE-RFIRSTDATE)+1,1+2*I)-R(STARTDATE-RFIRSTDATE),1+2*
2700 NEXT I
2710 GOSUB 3500
2720 PRINT "Do you want to do another calculation (<Y> or <N>):";PROCEED$
2730 WEND
2740 ERASE ACCUM : PRINT
2750 RETURN
2760 ' **** Print Accumulations ****
2770 IF LINECOUNT=0 THEN GOSUB 5500
2780 INPUT "Do you want to do another calculation (<Y> or <N>);PROCEED$
Appendix 1. Continued

October 1986

HIGLEY ET AL.: DEGDAY AND DEGREE-DAY ASSUMPTIONS

1011

1000 LPRINT : LPRINT : LPRINT : LINECOUNT=LINECOUNT+4
1010 LPRINT " Rectangle Triangle Sine Wave Sine Wave-Cooling"
1020 LPRINT " Dates ACDD AFDD ACDD AFDD ACDD AFDD ACDD AFDD"
1030 LPRINT " LINECOUNT=LINECOUNT+2
1040 LPRINT USING "#### #### #### #### #### #### #### #### ########"
1050 LPRINT "StartDate,StopDate,ACDD(AFDD),ACDD(AFDD),ACDD(AFDD),ACDD(AFDD),ACDD(AFDD),ACDD(AFDD),ACDD(AFDD),ACDD(AFDD),ACDD(AFDD)
1060 LPRINT USING "#### #### #### #### #### #### #### #### ########"
1070 LINECOUNT=LINECOUNT+2
1080 LPRINT USING "##################################################
1090 IF (LINECOUNT+4) MOD 66 =0 THEN LPRINT : LPRINT : LPRINT TAB( 40) LPRINT USING "##################################################"
1100 LINECOUNT=LINECOUNT+5
1110 RETURN

3000

3010 *** Access a results file ***
3020 PRINTCHR$(26) PRINT " Access Results" : PRINT
3030 PRINT "WARNING: Accessing a results file deletes results currently in machine memory."
3040 PRINT " (Disk files are not affected.)" : PRINT
3050 INPUT "Do you want to proceed «Y> or N)";PROCEED$ : PRINT
3060 IF PROCEED$="N" THEN RETURN
3070 INPUT "Enter name of the results file to be accessed: ",RFILENAME$
3080 PRINT : PRINT " Accessing Results"
3090 ON ERROR GOTO 6000
3100 OPEN "I",l,RFILENAME$
3110 ON ERROR GOTO 0
3120 ERASE R : DIM R(366,17)
3130 INPUT#l, TUP,TLO,C,RFIRSTDATE
3140 FOR I=1 TO 366
3150 FOR J=1 TO 17
3160 IF (I=1 AND J=1) THEN R(I,J)=RFIRSTDATE J=2
3170 FOR J=1 TO 17
3180 IF (I=1 AND J=1) THEN R(I,J)=RFIRSTDATE : J=2
3190 INPUT#l,R(I,J)
3200 RLASTDATE=R(I,1)
3210 NEXT J
3220 IF EOF(l) THEN I=366
3230 NEXT I
3240 CLOSE#1
3250 PRINT : PRINT " Printing subroutine" : PRINT
3260 PRINT "Enter title of output (80 characters maximum, do not use commas)"
3270 INPUT TITLES
3280 PRINT : PRINT " Printing in progress"
3290 N=40-LEN(TITLES)/2
3300 GOSUB 5500
3310 FOR I=1 TO (RLASTDATE-RFIRSTDATE)+1
3320 IF LINECOUNT<>12 THEN IF ((LINECOUNT+65) MOD 66)>0 THEN GOTO 5130 ELSE LPRINT : LPRINT : LPRINT : LINECOUNT=LINECOUNT+4
3330 LPRINT " Rectangle Triangle Sine Wave Sine Wave-Cooling"
3340 LPRINT " Date CDD/ACDD FDD/AFDD CDD/ACDD FDD/AFDD CDD/ACDD FDD/AFDD CDD/ACDD FDD/AFDD"
3350 LPRINT " LINECOUNT=LINECOUNT+2
3360 LPRINT USING "#### #### #### #### #### #### #### #### ########"
Appendix 1. Continued

```plaintext
*** Initial Output Heading ***
```

```plaintext
*** Error Trapping ***
```

```plaintext
*** exit to operating system ***
```

Appendix 2. DDINSTR, Program with Instructions for DEGDAY

7000 PRINT CHR$(26)
7010 PRINT " INSTRUCTIONS"
7020 PRINT " DEGDAY calculates degree-days using three algorithms: the rectangle"
7030 PRINT " method, the triangle (or trapezoid) method, and the sine wave method."
7040 PRINT "The sine wave method also is used to calculate cooling degree-days."
7050 PRINT "Calculations are made on a day-by-day basis, and results can be printed or"
7060 PRINT "stored on disk. Additionally, accumulations between dates can be calculated."
7070 PRINT " DEGDAY was developed to provide an easy method for making many degree-day"
7080 PRINT "calculations from a given data set. Thus, one data set for a given location"
7090 PRINT "can be used to determine the degree-days for many organisms with different"
7100 PRINT "developmental thresholds and for different starting and stopping dates. One"
7110 PRINT "important caution regarding DEGDAY, and the use of degree-days in general,"
7120 PRINT "is the recognition that degree-days are not the only determinant of"
7130 PRINT "physiological development for poikilotherms. Moreover, the use of developmental"
7140 PRINT "thresholds based on laboratory studies, and the use of temperature data that"
7150 PRINT "may not exactly represent developmental conditions, require that calculated"
7160 PRINT "degree-days be interpreted as estimates, not exact values. The following"
7170 PRINT "screens include instructions for using DEGDAY."
7180 PRINT " DEGDAY Requirements"
7190 PRINT " All responses to prompts by DEGDAY must be in capital letters, other than"
7200 PRINT "output titles and filenames. Responses in <> are the default responses and"
7210 PRINT "can be entered by hitting the return key. For example, you could answer no to"
7220 PRINT "the question ‘Do you want to print the results file (Y or <N>)?’ by entering"
7230 PRINT "‘N’ or just hitting the return key. Titles are restricted to 60 characters"
7240 PRINT "and cannot include commas. DEGDAY requires a printer with at least 60 columns"
7250 PRINT "and 66 lines per page. All input and output data filenames should be specified"
7260 PRINT "exactly as desired, including upper and lower case letters (if necessary) and"
7270 PRINT "a drive specifier (again, if necessary). For example, filenames such as"
7280 PRINT "DATA62, B:Input, A:RESULTS.DAT, etc, are appropriate."
7290 PRINT " Data for DEGDAY must be of the form:"
7300 PRINT " J"ULIAN DATE, DAILY MAXIMUM, DAILY MINIMUM"
7310 PRINT "Correct data format: Incorrect data format:"
Appendix 2. Continued

1014 FORUM: ENVIRONMENTAL ENTOMOLOGY Vol. 15, no. 5

Appendix 2. Continued

7540 PRINT
7550 PRINT "100,-10,-7"
7560 PRINT "100,-10,-7"
7570 PRINT
7580 INPUT "Hit <RETURN> to continue", N$ : PRINT CHR$(26)
7590 PRINT "Data Set Requirements Continued..."
7600 PRINT
7610 PRINT "7"
7620 PRINT "All Julian dates in a data file must be in sequence and only differ by"
7630 PRINT "one. If the dates in the data file cross years (as when calculating cooling"
7640 PRINT "degree-days for an entire winter), the Julian dates in the data file must be""
7650 PRINT "sequential. Thus 31 Dec. would be day 365 (366 for leap year), 1 Jan. would be"
7660 PRINT "entered as 366 (367 for leap year), 2 Jan. entered as 367 (368), etc."
7670 PRINT
7680 PRINT "7"
7690 PRINT "The maximum size for a DEGDAY input data file is 367 entries, but DEGDAY"
7700 PRINT "can only calculate degree-days for a maximum of 366 days. Some of the"
7710 PRINT "calculation techniques used in DEGDAY require temperatures from the next day"
7720 PRINT "to calculate degree-days for a given day. Thus, degree-days can only be"
7730 PRINT "calculated for one less than the number of days in the data file. For example,"
7740 PRINT "if a data set goes from Julian date 1 to 365, then degree-days only can be""
7750 PRINT "determined for a maximum of dates 1 to 364 (to calculate degree-days for date"
7760 PRINT "365 requires temperature data for date 366)."
7770 PRINT
7780 PRINT "7"
7790 PRINT "All data sets must be in ASCII format without higher order bits set. For"
7800 PRINT "details on data set requirements please review the Microsoft BASIC manual."
7810 PRINT "DEGDAY does not include routines for creating data sets because data sets can""
7820 PRINT "be most easily created and edited with available programs such as dBase II or"
7830 PRINT "Wordstar (in the nondocument mode)."
7840 PRINT
7850 PRINT "7"
7860 PRINT "Each choice prompts for the required information and includes some"
7870 PRINT "error-trapping features to prevent the program crashing with incorrect "
7880 PRINT "responses. As mentioned above, data sets must be created before using DEGDAY."
7890 PRINT
7900 PRINT "1. Calculate Degree-days (heating and cooling)"
7910 PRINT "This is the main course of the menu. You will be asked for a data file"
7920 PRINT "name (for the file containing temperature data). You will also be asked for a "
7930 PRINT "developmental minimum, a developmental maximum (if desired), and the starting "
7940 PRINT "and stopping dates. After calculations are completed you can print results,"
7950 PRINT "store results in a file (for accessing later), and you can calculate "
7960 PRINT "accumulations between dates."
7970 PRINT
7980 PRINT "2. Access Results File"
7990 PRINT "This option lets you access a previously stored results file. It reads"
8000 PRINT "the results file into machine memory so the file can then be printed or used "
8010 PRINT "to calculate accumulations between dates (in options 3 and 4)."
8020 PRINT
8030 PRINT "3. Print Results File"
8040 PRINT "This option lets you print a previously stored results file."
8050 PRINT
8060 INPUT "Hit <RETURN> to continue", N$ : PRINT CHR$(26)
8070 PRINT "Menu Choices continued..."
8080 PRINT
8090 PRINT "4. Calculate Degree-days Accumulated Between Dates"
Appendix 2. Continued

8100 PRINT "This option lets you calculate accumulations between dates for a"
8110 PRINT "previously stored results file. As many pairs of dates may be used as desired."
8120 PRINT
8130 PRINT "5. Instructions"
8135 PRINT "This option displays the material you are currently reading."
8140 PRINT
8150 PRINT "6. Exit to Operating System"
8155 PRINT "This option leaves the DEGDAY program."
8160 PRINT
8170 INPUT "Hit <RETURN> to continue",N$: PRINT CHRS(26)
8180 PRINT "HELP"
8190 PRINT
8200 PRINT "Hopefully, this is a section you won’t need to read. As previously"
8210 PRINT "mentioned, DEGDAY includes some modest error-trapping features to catch the"
8220 PRINT "most frequent errors (such as entering a nonexistent filename). The program"
8230 PRINT "has been tested and debugged (not that that’s any comfort if you can’t get"
8240 PRINT "it to work), and if you do have a problem you should first check your data set"
8250 PRINT "and your input. In particular, make certain that all the data set requirements"
8260 PRINT "described in the previous screens are met (this is by far the most probable"
8270 PRINT "source of the error). Specifically, make certain the data set begins and ends"
8280 PRINT "with data points (not blank lines), uses Julian dates followed by maximum and"
8290 PRINT "minimum temperatures, and has negative signs adjacent to numbers."
8300 PRINT "Review information in these instructions, and for more detailed information on"
8310 PRINT "data set requirements for BASIC programs is available in the Microsoft BASIC"
8320 PRINT "Handbook."
8330 PRINT
8340 INPUT "Hit <RETURN> to continue",N$: PRINT CHRS(26)
8350 PRINT
8360 PRINT "The formulas used in DEGDAY were derived from the following sources:"
8370 PRINT
8380 PRINT "Rectangle Method"
8390 PRINT "Arnold, C. Y. 1960. Maximum-minimum temperatures as a basis for"
8410 PRINT "Triangle Method"
8430 PRINT "to a basic thermal summation model for predicting the time of"
8440 PRINT "emergence of the adult western cherry fruit fly, Rhagoletis"
8450 PRINT "indifferens Curran (Dipt., Tephritidae). Z. Angew. Entomol."
8460 PRINT "94: 401-407."
8470 PRINT "Sine Wave Method"
8490 PRINT "days. Environ. Entomol. 5: 380-396."
8500 PRINT
8510 PRINT "Another important reference regarding degree-day calculations and their"
8520 PRINT "appropriate use is:"n
8540 PRINT "Entomol. 12:613-619."
8550 PRINT
8560 INPUT "Hit <RETURN> to return to DEGDAY menu",N$
8570 CHAIN "DEGDAY",400
Appendix 3. Alternative Code for Degree-day Calculations by Whole Day

```
1660 ' **** Triangle and Sine Wave Methods
1670 ' TMAX=D(I,2) : Tmin=D(I,3)
1680 ' **** test for cases 1-6
1690 ' A=(TMAX-TMIN)/2 : MT=(TMAX+TMIN)/2
1700 IF TMIN>TUP AND TMAX<TUP GOTO 1770 ' ** Case 1
1710 IF TMIN<TLO AND TMAX>TLO GOTO 1820 ' ** Case 2
1720 IF TMIN>TLO AND TMAX<TLO GOTO 1870 ' ** Case 3
1730 IF TMIN<TLO AND TMAX<TUP GOTO 1920 ' ** Case 4
1740 IF TMIN>TLO AND TMAX>TUP GOTO 2040 ' ** Case 5
1750 IF TMIN>TLO AND TMAX>TUP GOTO 2040 ' ** Case 6
1760 ' **** Case 1 ****
1770 TRI(C,1)=(TUP-TLO)
1780 SINE(C,1)=(TUP-TLO)
1800 COOL(C,1)=0
1810 GOTO 2160
1820 ' **** Case 2 ****
1830 TRI(C,1)=0
1840 SINE(C,1)=0
1850 COOL(C,1)=(TLO-A)
1860 GOTO 2160
1870 ' **** Case 3 ****
1880 TRI(C,1)=(MT-TLO)
1890 SINE(C,1)=(MT-TLO)
1900 COOL(C,1)=0
1910 GOTO 2160
1920 ' **** Case 4 ****
1930 TRI(C,1)=(TMAX-TL0)*2/(TMAX-TMIN)*2)
1940 X1=(TLO-MT)/A
1950 T1=ATN(X1/SQR(1-X1^2))
1960 T2=1.5708
1970 GOTO 2100
1980 ' **** Case 5 ****
1990 TRI(C,1)=(MT-TLO)-(TMAX-TUP)*2/(TMAX-TMIN)*2)
2000 X2=(TUP-MT)/A
2010 T1=-1.5708
2020 T2=ATN(X2/SQR(1-X2^2))
2030 GOTO 2100 ' ** goto accumulation formula **
2040 ' **** Case 6 ****
2050 TRI(C,1)=(TMAX-TL0)*2-(TMAX-TUP)*2)/(TMAX-TMIN)*2)
2060 X1=(TLO-MT)/A
2070 X2=(TUP-MT)/A
2080 T1=ATN(X1/SQR(1-X1^2))
2090 T2=ATN(X2/SQR(1-X2^2))
2100 ' **** Calculation formula for cases 4-6 ****
2110 SINE(C,1)=.31831*(TLO-MT)*(T2-T1)+A*COS(T1)-COS(T2)+TUP-TLO)*
2120 *(1.5708-T2))
2130 COOL(C,1)=.31831*(TLO-MT)*(T1+1.5708)+A*COS(T1))
```