UNIVERSITY OF MINNESOTA
ST. ANTHONY FALLS LABORATORY
Engineering, Environmental and Geophysical Fluid Dynamics

Project Report 382

Correlations of Minnesota Stream Water Temperatures with Air Temperatures

by

John M. Pilgrim, Xing Fang and

Heinz G. Stefan

Prepared for

NATIONAL AGRICULTURAL WATER QUALITY LABORATORY
Agricultural Research Service
U. S. Department of Agriculture
Durant, Oklahoma

December 1995
Minneapolis, Minnesota

Abstract

Air temperatures are sometimes used as substitutes for stream temperatures. To examine the errors associated with this substitution, linear relationships between 43 Minnesota stream water temperature records and associated air temperature records were analyzed. Weather monitoring stations were, on average, 23.3 miles from the stream stations. From the lumped data set (38,082 data pairs), the general equations, T_w=4.4+0.81T_a, T_w=1.9+0.97$T_{a,}$, T_w=0.7+1.04T_a, and T_w=3.3+0.89T_a ,with temperatures in °C, were derived for daily, weekly, monthly, and yearly mean temperatures, respectively. Standard deviations between all measured and predicted water temperatures were 3.5°C (daily), 2.6°C (weekly), 1.9°C (monthly), and 1.3°C (yearly). Separate analyses for each specific stream gave lower standard deviations. The measured water temperatures follow the annual air temperature cycle closely. No time lags were taken into account, and periods of ice cover were excluded from the analysis.

(KEY TERMS: stream temperature; water quality; simulation; modeling/statistics; meteorology/ climatology; hydrobiology)

Acknowledgements

This study was funded in part by the USEPA and the USDA/ARS. Robert Schoeller from the USEPA/ERLD provided the USGS stream temperature data records.

Table of Contents

List of Figures

List of Tables

Introduction

As part of a growing public interest in environmental issues, projected global climate change has received much attention in recent years. Although studies in this area have focused primarily on land, oceans, and the atmosphere, there is also a need to study projected global change effects on streams and lakes. Accurate forecasting of temperatures in streams is important for the prediction of how fish and other aquatic organisms will be affected by these projected climate changes. Stream water temperature data are more difficult to obtain than air temperatures, and air temperature is therefore often substituted for water temperature. In reality, however, air temperature offers only a rough approximation of stream water temperature, and therefore some type of correlation analysis and error assessment is worthwhile if not necessary.

Air temperature is not the only physical parameter influencing water temperature. Other parameters of influence are solar radiation, relative humidity, wind speed, water depth, groundwater inflow artificial heat inputs, and thermal conductivity of the sediments. Several numerical heat budget models of varying complexities have been developed to predict stream water temperatures (Brown, 1972; Jobson, 1973; Novotny and Krenkel, 1973; Noble, 1979; Stefan et al., 1980; Paily and Macagno, 1976; Morin et al., 1987; and others). Meteorological data required for such models are often not available, and the effort necessary to acquire them can be substantial. It is therefore useful to have a simple, approximate relationship between daily, weekly, or monthly air and water temperatures.

In a previous, closely related study, Stefan and Preud'homme (1993) investigated the water/air temperature relationships of 11 streams in Minnesota, Louisiana, Arkansas, and Texas. The study used a linear relationship of the form

$$T_w(t) = A + B \bullet T_{a(t)} \qquad (1)$$

proposed originally by Smith (1981) for British rivers, where $T_w(t)$ and $T_a(t)$ are the stream water and air temperatures at time t, and A and B are constants. Stefan and Preud'homme also made use of a time lag, ranging from four hours to seven days depending on the depth of the stream, to improve temperature predictions. They determined that introducing a time lag into the correlation between daily air and water temperatures had an effect only for major rivers; furthermore, the time lag did not improve the predictions by much. The greatest improvement in the standard error of prediction was from 4.42°C to 3.96°C when a 5-day time lag was included for the longest river they investigated (Mississippi River). The study also found that the correlations became markedly better when weekly average temperatures were used instead of daily values.

The primary objective of this study is to investigate the accuracy of the simple linear model (Eq. 1) on a broader assortment of streams and time scales. This study uses daily, weekly, monthly, and yearly time scales, and applies them to data sets from 43 streams scattered throughout the state of Minnesota. Applying the model to many streams and several time scales will provide additional insight into predictions of stream temperature. Of particular interest is the determination of the time scale which produces the most accurate temperature predictions. This only becomes possible when a sufficiently large data set is used for correlation.

This paper is limited to streams, i.e. water bodies which are well-mixed in the vertical and transverse direction of a cross-section. It is also limited to the open water period from April 1 to October 31, when streams in the northern United States are usually ice free. In ice-covered streams, the temperature of the water is usually at or slightly above 0°C regardless of air temperature. No tests are done to determine the significance and effect of serial correlation on the air/water temperature correlation.

Air and Water Temperature Database

For this study, temperature records for 43 streams scattered throughout the state of Minnesota were obtained from the U.S. Geological Survey (USGS) stream database through the U.S. Environmental Protection Agency (USEPA) in Duluth, Minn. Of these 43 records, two proved to be unusable. One of these two, from the Pigeon River near Grand Portage, Minn., contained only readings taking during morning hours or at random hours during the day. Stefan and Preud'homme (1993) showed that since temperature can fluctuate significantly over the course of a day, random temperature readings are imprecise in general. Furthermore, morning temperature readings are not representative of the daily mean temperature. The second deleted record, from the Mississippi River at Winona, Minn., contained readings for only a few days within the acceptable period of study.

The remaining 41 records consisted of either true daily mean temperature readings or daily maximum/minimum temperature pairs, from which mean readings can be safely approximated (Stefan and Preud'homme, 1993). This study is also limited to the period from April 1 to October 31, when Minnesota streams are usually free of ice cover.

Air temperature records were obtained from the Midwest Climate Center of the Illinois State Water Survey in Champaign, Ill. For each stream in this study, corresponding air temperature readings were taken from the record of the nearest available monitoring station. Some air temperature stations considered for this study were eliminated because of large gaps in the temperature records. The records for the stations found to be acceptable had few missing data points, and in every case the true daily mean temperature was recorded. The locations of the stream temperature and air temperature monitoring stations are summarized in Table 1. The distance between water and air monitoring stations is given in the last column of Table 1. Temperature records are detailed in Table 2.

Table 1. Water and air temperature monitoring stations.

Stream Station	Location * Lat.	Long.	Air Temp. Station	Location * Lat.	Long.	Distance (mi)
1. PIGEON RIVER (near Grand Portage, MN)	48 01	89 37	Grand Marais, MN	47 44	90 21	52
2. BAPTISM RIVER (near Beaver Bay, MN)	47 20	91 12	Two Harbors, MN	47 01	91 40	35
3. PARTRIDGE RIVER (at Hoyt Lakes, MN)	47 32	92 07	Babbitt, MN	47 41	91 55	16
4. PARTRIDGE RIVER (near Aurora, MN)	47 31	92 11	Babbitt, MN	47 41	91 55	20
5. ST. LOUIS RIVER (at Scanlon, MN)	46 42	92 25	Cloquet, MN	46 42	92 31	7
6. WILD RICE RIVER (at Twin Valley, MN)	47 16	96 37	Crookston, MN	47 48	96 37	25
7. RED LAKE RIVER (at Crookston, MN)	47 47	96 37	Crookston, MN	47 48	96 37	1
8. RED RIVER OF THE NORTH (at Oslo, MN)	48 12	97 08	Argyle, MN	48 20	96 44	29
9. RED RIVER OF THE NORTH (at Emerson, MN)	49 00	97 13	Hallock, MN	48 46	96 57	21
10. ROSEAU RIVER (near Caribou, MN)	48 59	96 28	Hallock, MN	48 46	96 57	35
11. KAWISHIWI RIVER (near Ely, MN)	47 55	91 32	Winton, MN Power Plant	47 56	91 46	16
12. STONY RIVER (near Babbitt, MN)	47 42	91 46	Babbitt, MN	47 41	91 55	11
13. DUNKA RIVER (near Babbitt, MN)	47 42	91 52	Babbitt, MN	47 41	91 55	3
14. LITTLE FORK RIVER (near Littlefork, MN)	48 24	93 33	Babbitt, MN	47 41	91 55	118
15. RAINY RIVER (at Manitou Rapids, MN)	48 38	93 55	Babbitt, MN	47 41	91 55	145
16. MISSISSIPPI RIVER (near Royalton, MN)	45 52	94 21	Melrose, MN	45 40	94 49	33
17. ELK RIVER (near Big Lake, MN)	45 20	93 40	Minn.-St. Paul Airport	44 53	93 13	38
18. MISSISSIPPI RIVER (near Anoka, MN)	45 08	93 18	Minn.-St. Paul Airport	44 53	93 13	13
19. MISSISSIPPI RIVER (at Fridley, MN)	45 06	93 17	Minn.-St. Paul Airport	44 53	93 13	11
20. MISSISSIPPI RIVER (at Ford Plant)	44 55	93 12	Saint Paul, MN	44 58	93 05	8
21. YELLOW BANK RIVER (near Odessa, MN)	45 14	96 21	Milan, MN	45 08	95 56	29

Table 1. Water and air temperature monitoring stations.(Cont'd)

Stream Station	Location * Lat.	Location * Long.	Air Temp. Station	Location * Lat.	Location * Long.	Distance (mi)
22. CANBY CREEK (near Canby, MN)	44 41	96 21	Canby, MN	44 43	96 17	5
23. MINNESOTA RIVER (at New Ulm, MN)	44 19	94 27	Jordan, MN	44 39	93 37	60
24. MINNESOTA RIVER (at Mankato, MN)	44 10	94 00	Jordan, MN	44 39	93 37	35
25. MINNESOTA RIVER (near Jordan, MN)	44 42	93 39	Jordan, MN	44 39	93 37	3
26. MINNESOTA RIVER (at Burnsville, MN)	44 49	93 15	Minn.-St. Paul Airport	44 53	93 13	4
27. MINNESOTA RIVER (at Fort Snelling St. Pk.)	44 52	93 12	Saint Paul, MN	44 58	93 05	9
28. MISSISSIPPI RIVER (at Saint Paul, MN)	44 57	93 05	Saint Paul, MN	44 58	93 05	1
29. MISSISSIPPI RIVER (at Industrial Molasses)	44 56	93 03	Saint Paul, MN	44 58	93 05	3
30. MISSISSIPPI RIVER (at Newport, MN)	44 52	93 00	Saint Paul, MN	44 58	93 05	7
31. MISSISSIPPI RIVER (at Cottage Grove, MN)	44 48	93 01	Saint Paul, MN	44 58	93 05	9
32. MISSISSIPPI RIVER (at L&D #2, Hastings, MN)	44 46	92 52	Saint Paul, MN	44 58	93 05	18
33. MISSISSIPPI RIVER (below L&D #2, Hastings, MN)	44 45	92 51	Saint Paul, MN	44 58	93 05	19
34. ST. CROIX RIVER (at Afton, MN)	44 54	92 47	Saint Paul, MN	44 58	93 05	21
35. MISSISSIPPI RIVER (near Red Wing, MN)	44 37	92 37	Saint Paul, MN	44 58	93 05	37
36. VERMILLION RIVER (near Empire, MN)	44 40	93 03	Farmington, MN	44 40	93 11	9
37. ZUMBRO RIVER (at Kellogg, MN)	44 19	92 00	Winona, MN	44 03	91 38	28
38. WHITEWATER RIVER (near Beaver, MN)	44 09	92 00	Winona, MN	44 03	91 38	26
39. MISSISSIPPI RIVER (at Winona, MN)	44 03	91 38	Winona, MN	44 03	91 38	0
40. ROOT RIVER (near Houston, MN)	43 46	91 34	Winona, MN	44 03	91 38	14
41. SOUTH FORK ROOT RIVER (near Houston, MN)	43 44	91 34	Winona, MN	44 03	91 38	16
42. 060 N12 W03 DCA EM8 (near Babbitt, MN)	47 42	91 51	Babbitt, MN	47 41	91 55	5
43. EM-3 (near Babbitt, MN)	47 43	91 50	Babbitt, MN	47 41	91 55	7

* Location given in degrees and minutes of northern latitude and longitude

Table 2. Water temperature records

Stream Station	Period	Water Temperature Record Format	Length (yrs)
1. PIGEON RIVER (near Grand Portage, MN)	1970-1973	Daily morning, random	4
2. BAPTISM RIVER (near Beaver Bay, MN)	1970-1973, 1981-1983	Daily evening Daily max, min, mean	7
3. PARTRIDGE RIVER (at Hoyt Lakes, MN)	1976-1985	Daily max, min, mean	10
4. PARTRIDGE RIVER (near Aurora, MN)	1956-1963, 1975-1976	Daily evening, random Daily max, min, mean	10
5. ST. LOUIS RIVER (at Scanlon, MN)	1980-1983	Daily max, min, mean	4
6. WILD RICE RIVER (at Twin Valley, MN)	1973, 1976-1979	Daily random Daily max, min, mean	5
7. RED LAKE RIVER (at Crookston, MN)	1980-1983	Daily max, min, mean	4
8. RED RIVER OF THE NORTH (at Oslo, MN)	1974-1978	Daily max, min	5
9. RED RIVER OF THE NORTH (at Emerson, MN)	1990-1991	Daily mean	2
10. ROSEAU RIVER (near Caribou, MN)	1980-1983	Daily max, min, mean	4
11. KAWISHIWI RIVER (near Ely, MN)	1966-1978, 1979-1981	Daily max, min Daily max, min, mean	16
12. STONY RIVER (near Babbitt, MN)	1975-1976	Daily max, min, mean	2
13. DUNKA RIVER (near Babbitt, MN)	1975-1980	Daily max, min, mean	6
14. LITTLE FORK RIVER (near Littlefork, MN)	1973, 1980-1983	Daily random Daily max, min, mean	5
15. RAINY RIVER (at Manitou Rapids, MN)	1980-1983	Daily max, min, mean	4
16. MISSISSIPPI RIVER (near Royalton, MN)	1980-1983	Daily max, min, mean	4
17. ELK RIVER (near Big Lake, MN)	1977-1980, 1981	Daily random Daily max, min, mean	5
18. MISSISSIPPI RIVER (near Anoka, MN)	1960-1963, 1976-1984, 1985-1986, 1988-1991	Daily random Daily max, min, mean, random Daily random	19
19. MISSISSIPPI RIVER (at Fridley, MN)	1974-1986	Daily max, min, mean	13
20. MISSISSIPPI RIVER (at Ford Plant)	1973-1978, 1980-1981	Daily max, min, mean	8
21. YELLOW BANK RIVER (near Odessa, MN)	1973-1974, 1975, 1976-1986	Daily random Daily mean Daily random	14
22. CANBY CREEK (near Canby, MN)	1980-1981	Daily max, min, mean	2

Table 2. Water temperature records (Cont'd)

Stream Station	Period	Water Temperature Record Format	Length (yrs)
23. MINNESOTA RIVER (at New Ulm, MN)	1968-1974, 1975, 1976	Daily random Daily mean Daily random	9
24. MINNESOTA RIVER (at Mankato, MN)	1968-1972, 1973-1976, 1977-1986, 1990-1991	Daily random Daily mean, random Daily random	21
25. MINNESOTA RIVER (near Jordan, MN)	1973-1986	Daily max, min, mean	14
26. MINNESOTA RIVER (at Burnsville, MN)	1980-1982	Daily max, min, mean	3
27. MINNESOTA RIVER (at Fort Snelling St. Pk.)	1978-1982	Daily max, min, mean	5
28. MISSISSIPPI RIVER (at Saint Paul, MN)	1956-1976, 1977-1987	Daily max, min Daily mean	32
29. MISSISSIPPI RIVER (at Industrial Molasses)	1976-1986	Daily max, min, mean	11
30. MISSISSIPPI RIVER (at Newport, MN)	1978-1988	Daily max, min, mean	11
31. MISSISSIPPI RIVER (at Cottage Grove, MN)	1977-1988	Daily max, min, mean	12
32. MISSISSIPPI RIVER (at L&D #2, Hastings, MN)	1973-1988	Daily max, min, mean	16
33. MISSISSIPPI RIVER (below L&D #2, Hastings, MN)	1977	Daily max, min, mean	1
34. ST. CROIX RIVER (at Afton, MN)	1976-1982	Daily max, min, mean	7
35. MISSISSIPPI RIVER (near Red Wing, MN)	1969-1977, 1978-1982	Daily max, min Daily mean	14
36. VERMILLION RIVER (near Empire, MN)	1974-1988	Daily max, min, mean	15
37. ZUMBRO RIVER (at Kellogg, MN)	1975-1976, 1977-1981	Daily mean, random Daily random	7
38. WHITEWATER RIVER (near Beaver, MN)	1975-1976, 1977-1981	Daily mean, random Daily random	7
39. MISSISSIPPI RIVER (at Winona, MN)	1974-1975, 1976-1986	Daily mean Daily random	13
40. ROOT RIVER (near Houston, MN)	1968-1974, 1975, 1976-1981	Daily random Daily mean Daily random	14
41. SOUTH FORK ROOT RIVER (near Houston, MN)	1975-1976, 1977-1981	Daily mean, random Daily random	7
42. 060 N12 W03 DCA EM8 (near Babbitt, MN)	1976	Daily max, min, mean	1
43. EM-3 (near Babbitt, MN)	1977	Daily max, min, mean	1

Methods and Techniques

Before generating the linear correlations based on Eq. 1, a significant amount of data manipulation and reformatting was necessary. This was accomplished using three Fortran computer programs which are listed and described in detail in the Appendix.

To carry out the correlations, it was first necessary to determine the mean temperatures for each of the time scales: daily, weekly, monthly, and yearly. Daily mean temperatures were directly extracted from the water and air temperature records if available, or calculated by averaging the maximum and minimum temperature readings reported for each day. The daily mean temperatures were then averaged over each week, month, and year to generate the mean weekly, monthly, and yearly temperatures. These mean temperatures were accepted only if daily temperatures were present for the majority of the days in the time period (at least 4 days per week, at least 16 days per month, or at least 107 days per year between April and October). Data that did not fulfill this criterion were ignored as were a few readings that fell below 0°C, where ice cover would still have been present.

Using the mean stream water and air temperatures, correlations for each stream and time scale were developed by calculating the constants, A and B in Eq. 1, using least-squares regression. These correlations were only performed if there were at least 4 data points available to correlate for a given stream and time scale. Then, the measured stream temperatures were compared to the temperatures predicted by the model, and from this, statistical parameters were calculated to determine the quality of each correlation. The correlation coefficient, R^2, is related to the ratio of the variance between field data and model predictions to the variance in the field data. It is defined as

$$R^2 = 1 - \frac{S_p^2}{S_e^2} \tag{2}$$

where S_p is the standard error of prediction and S_e is the standard deviation of the data from the mean. S_p and S_e are defined as

$$S_p = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}} \tag{3}$$

and

$$S_e = \sqrt{\frac{\sum_{i=1}^{n}\left(y_i - \overline{y}\right)^2}{n}} \tag{4}$$

where y_i is one of n measured water temperatures, x_i represents a water temperature estimated from Eq. 1, and $\overline{y}$ is the mean of water temperature field data, given as

$$\overline{y} = \frac{\sum_{i=1}^{n} y_i}{n} \tag{5}$$

Both R^2 and S_p are measures of the quality of a correlation. The correlation objective is to have R^2 approach 1 and S_p approach 0.

Finally, a lumped data correlation analysis was performed using the data from of all the streams simultaneously. Again, the constants A and B were calculated along with the R^2, S_p, and S_e values. Having completed the analysis, the correlations for each stream on daily, weekly, monthly, and yearly time scales, as well as the lumped correlation for the same time scales, were summarized in a computer file.

Results

The analysis produced the correlation results presented in Tables 3 to 7. In each table, A and B are the coefficients in Eq. 1, and N_c indicates the number of data points that were actually correlated. The data for each stream had distinct beginning and ending dates, and N_{TOT} indicates the total number of data points possible if the data were complete within the given time period. Thus, the difference between N_c and N_{TOT} gives the number of missing or eliminated data points, and the ratio N_c / N_{TOT} illustrates the completeness of each data set. The correlation coefficient R^2 and the standard error of prediction S_p, also given in Tables 3 to 7, indicate the quality of the correlations. The closer that R^2 is to 1 and S_p is to 0, the better the fit.

Individual Streams

To visually demonstrate the effectiveness of these correlations, an example is provided in Fig. 1 for the Red River of the North at Oslo, Minn. The Red River of the North (No. 8 in Table 1) has both a reasonable correlation between stream and air temperatures and a reasonably complete set of data for all time scales. Fig. 1 shows all of the water and air temperature readings along with the regression lines (appearing as solid lines) for each time scale. The dashed lines indicate where a one-to-one correlation ($T_w = T_a$) would lie. The data correlate well for weekly, monthly, and yearly time scales, with correlation coefficients R^2 of 0.86, 0.90, and 0.97 respectively, but not as well for daily values (0.69). The weekly, monthly, and yearly regression lines have slopes that are nearly 1 (1.04, 1.07, and 1.05 respectively), but each line is offset by a constant at zero air temperature. This is typical of most streams.

As the time scale lengthens, the data points become fewer. For example, the monthly and yearly regressions for the Red River of the North (No. 8) were calculated from only 19 and 4 data points, respectively. The fewer data points also cover a narrower range of temperatures around the mean for the open water season. The correlation equations should not be extended or applied to temperatures much higher or lower than the data points. However, the regression lines in Fig. 1 are drawn well beyond the data range simply to facilitate comparisons between the different time scales.

As indicated earlier, the records from two of the streams—Pigeon River near Grand Portage, Minn. (No. 1) and Mississippi River at Winona, Minn. (No. 39)—had insufficient data for correlation analysis. Additionally, two particular stations were found to give poor results after carrying out the analysis. Station No. 42, near Babbitt, Minn., exhibited a zero

Table 3. Results of daily water and air temperature analysis.

Stream Station	A (°C)	B	N_c	N_{TOT}	R^2	S_p (°C)	S_e (°C)
2. BAPTISM RIVER	0.99	0.913	542	604	0.775	2.80	5.90
3. PARTRIDGE RIVER	2.23	0.922	1567	2127	0.773	3.09	6.47
4. PARTRIDGE RIVER	3.81	0.578	25	26	0.639	1.74	2.90
5. ST. LOUIS RIVER	2.34	0.933	516	604	0.782	3.13	6.70
6. WILD RICE RIVER	2.38	0.953	530	786	0.790	3.19	6.98
7. RED LAKE RIVER	3.09	0.904	228	434	0.709	3.36	6.23
8. RED RIVER OF THE NORTH	5.26	0.816	608	804	0.694	3.30	5.96
9. RED RIVER OF THE NORTH	4.52	0.862	230	243	0.750	3.80	7.60
10. ROSEAU RIVER	2.82	0.870	420	638	0.815	2.83	6.58
11. KAWISHIWI RIVER	5.21	0.756	2544	3288	0.610	3.85	6.17
12. STONY RIVER	3.40	0.846	202	203	0.690	3.58	6.43
13. DUNKA RIVER	2.03	0.949	905	1038	0.797	3.00	6.65
14. LITTLE FORK RIVER	2.65	0.918	399	445	0.709	3.37	6.24
15. RAINY RIVER	2.14	0.888	442	472	0.742	3.16	6.22
16. MISSISSIPPI RIVER	4.06	0.865	597	618	0.768	3.04	6.31
17. ELK RIVER	4.80	0.787	21	178	0.808	1.54	3.52
18. MISSISSIPPI RIVER	3.17	0.861	562	1069	0.746	3.22	6.39
19. MISSISSIPPI RIVER	3.52	0.834	1818	2563	0.746	3.22	6.40
20. MISSISSIPPI RIVER	4.23	0.814	1136	1922	0.674	3.52	6.17
21. YELLOW BANK RIVER	0.32	0.956	198	214	0.828	2.92	7.04
22. CANBY CREEK	3.09	0.833	185	185	0.830	2.07	5.03
23. MINNESOTA RIVER	6.79	0.856	406	428	0.662	3.81	6.56
24. MINNESOTA RIVER	5.15	0.838	450	764	0.684	3.62	6.44
25. MINNESOTA RIVER	5.42	0.821	2070	2872	0.721	3.22	6.09
26. MINNESOTA RIVER	4.39	0.823	354	579	0.699	3.31	6.05
27. MINNESOTA RIVER	4.99	0.759	573	887	0.710	3.12	5.80
28. MISSISSIPPI RIVER	5.56	0.774	6174	6603	0.646	3.72	6.25
29. MISSISSIPPI RIVER	4.78	0.799	1577	2312	0.715	3.16	5.92
30. MISSISSIPPI RIVER	5.00	0.773	1669	2035	0.688	3.34	5.99
31. MISSISSIPPI RIVER	4.57	0.791	1786	2378	0.733	3.07	5.94
32. MISSISSIPPI RIVER	4.53	0.813	2102	3161	0.734	3.12	6.04
33. MISSISSIPPI RIVER	6.08	0.752	61	67	0.449	4.86	6.55
34. ST. CROIX RIVER	5.44	0.759	854	1276	0.611	3.78	6.06
35. MISSISSIPPI RIVER	5.16	0.807	2700	2869	0.722	3.27	6.20
36. VERMILLION RIVER	4.72	0.638	2472	3179	0.831	1.85	4.49
37. ZUMBRO RIVER	5.23	0.726	245	270	0.776	2.45	5.18
38. WHITEWATER RIVER	6.37	0.564	183	302	0.638	2.67	4.43
40. ROOT RIVER	5.92	0.789	366	406	0.824	2.39	5.69
41. SOUTH FORK ROOT RIVER	6.88	0.650	279	285	0.790	2.09	4.55
Average	4.18	0.815	974	1260	0.714	3.23	6.07

Table 4. Results of weekly water and air temperature analysis.

Stream Station	A (°C)	B	N_c	N_{TOT}	R^2	S_p (°C)	S_e (°C)
2. BAPTISM RIVER	–0.31	1.012	77	87	0.854	2.11	5.52
3. PARTRIDGE RIVER	0.01	1.085	221	307	0.893	2.07	6.33
5. ST. LOUIS RIVER	–0.24	1.111	74	87	0.915	1.96	6.71
6. WILD RICE RIVER	–0.67	1.145	77	113	0.917	2.04	7.06
7. RED LAKE RIVER	–0.05	1.140	32	63	0.921	1.72	6.12
8. RED RIVER OF THE NORTH	1.77	1.041	85	116	0.862	2.19	5.91
9. RED RIVER OF THE NORTH	2.48	1.005	34	35	0.861	2.79	7.51
10. ROSEAU RIVER	0.68	1.028	61	92	0.925	1.76	6.45
11. KAWISHIWI RIVER	2.99	0.920	364	476	0.745	3.13	6.20
12. STONY RIVER	0.74	1.028	29	29	0.812	2.78	6.43
13. DUNKA RIVER	–0.36	1.116	132	150	0.902	2.07	6.61
14. LITTLE FORK RIVER	–0.41	1.137	57	64	0.864	2.27	6.16
15. RAINY RIVER	–0.50	1.085	64	68	0.870	2.27	6.31
16. MISSISSIPPI RIVER	1.46	1.024	86	89	0.904	1.96	6.31
18. MISSISSIPPI RIVER	1.06	0.986	81	154	0.854	2.37	6.20
19. MISSISSIPPI RIVER	1.18	0.975	262	370	0.877	2.27	6.45
20. MISSISSIPPI RIVER	0.69	1.027	159	278	0.841	2.43	6.08
21. YELLOW BANK RIVER	–1.51	1.069	29	31	0.934	1.68	6.54
22. CANBY CREEK	0.13	0.999	26	26	0.960	0.95	4.74
23. MINNESOTA RIVER	3.35	1.070	59	62	0.813	2.82	6.53
24. MINNESOTA RIVER	2.49	1.004	65	110	0.821	2.63	6.21
25. MINNESOTA RIVER	2.65	0.994	294	415	0.860	2.25	6.01
26. MINNESOTA RIVER	0.76	1.048	49	83	0.877	2.10	6.00
27. MINNESOTA RIVER	2.34	0.919	82	128	0.857	2.19	5.79
28. MISSISSIPPI RIVER	2.40	0.963	898	956	0.806	2.78	6.30
29. MISSISSIPPI RIVER	1.72	0.976	224	334	0.868	2.17	5.98
30. MISSISSIPPI RIVER	1.96	0.950	242	294	0.853	2.31	6.02
31. MISSISSIPPI RIVER	1.76	0.951	252	344	0.879	2.05	5.88
32. MISSISSIPPI RIVER	1.68	0.970	301	457	0.860	2.24	5.98
33. MISSISSIPPI RIVER	2.39	0.983	9	9	0.665	3.71	6.41
34. ST. CROIX RIVER	2.47	0.935	122	184	0.753	3.02	6.08
35. MISSISSIPPI RIVER	2.43	0.970	391	415	0.859	2.35	6.25
36. VERMILLION RIVER	3.53	0.710	353	460	0.904	1.35	4.35
37. ZUMBRO RIVER	3.30	0.851	36	39	0.920	1.37	4.86
38. WHITEWATER RIVER	3.78	0.729	29	43	0.875	1.53	4.31
40. ROOT RIVER	4.01	0.908	52	58	0.960	1.10	5.47
41. SOUTH FORK ROOT RIVER	4.83	0.772	41	41	0.963	0.84	4.39
Average	1.65	0.990	147	191	0.852	2.30	6.04

Table 5. Results of monthly water and air temperature analysis.

Stream Station	A (°C)	B	N_c	N_{TOT}	R^2	S_p (°C)	S_e (°C)
2. BAPTISM RIVER	0.06	0.996	17	19	0.885	1.72	5.08
3. PARTRIDGE RIVER	–1.02	1.164	50	69	0.959	1.18	5.84
5. ST. LOUIS RIVER	–0.42	1.164	16	19	0.972	1.04	6.21
6. WILD RICE RIVER	–2.22	1.243	17	25	0.958	1.28	6.26
7. RED LAKE RIVER	0.10	1.148	7	14	0.947	1.04	4.53
8. RED RIVER OF THE NORTH	1.53	1.067	19	26	0.902	1.52	4.86
9. RED RIVER OF THE NORTH	1.02	1.108	8	8	0.970	1.22	7.03
10. ROSEAU RIVER	0.74	1.034	14	21	0.973	0.97	5.88
11. KAWISHIWI RIVER	1.38	1.043	84	107	0.842	2.32	5.84
12. STONY RIVER	–0.33	1.124	6	6	0.924	1.70	6.17
13. DUNKA RIVER	–2.05	1.230	30	34	0.957	1.27	6.15
14. LITTLE FORK RIVER	–1.24	1.205	13	14	0.953	1.23	5.70
15. RAINY RIVER	–0.42	1.098	14	15	0.947	1.31	5.68
16. MISSISSIPPI RIVER	1.25	1.046	19	20	0.966	1.05	5.71
18. MISSISSIPPI RIVER	–0.68	1.094	20	35	0.920	1.63	5.78
19. MISSISSIPPI RIVER	–0.10	1.054	58	84	0.925	1.59	5.79
20. MISSISSIPPI RIVER	–0.44	1.091	37	63	0.897	1.76	5.47
21. YELLOW BANK RIVER	–3.07	1.157	7	7	0.978	1.03	7.05
22. CANBY CREEK	–0.77	1.048	6	6	0.997	0.24	4.37
23. MINNESOTA RIVER	2.14	1.165	13	14	0.936	1.36	5.37
24. MINNESOTA RIVER	0.52	1.126	15	25	0.905	1.82	5.91
25. MINNESOTA RIVER	1.28	1.083	69	94	0.928	1.52	5.68
26. MINNESOTA RIVER	0.78	1.067	11	19	0.977	0.81	5.39
27. MINNESOTA RIVER	1.11	0.991	18	29	0.941	1.26	5.21
28. MISSISSIPPI RIVER	0.19	1.095	205	216	0.899	1.88	5.93
29. MISSISSIPPI RIVER	0.40	1.048	53	75	0.940	1.34	5.45
30. MISSISSIPPI RIVER	0.69	1.028	53	66	0.931	1.47	5.59
31. MISSISSIPPI RIVER	0.46	1.029	61	77	0.948	1.27	5.60
32. MISSISSIPPI RIVER	0.37	1.053	68	103	0.944	1.33	5.62
34. ST. CROIX RIVER	0.21	1.067	28	42	0.852	2.23	5.79
35. MISSISSIPPI RIVER	0.42	1.089	91	94	0.937	1.48	5.88
36. VERMILLION RIVER	3.10	0.738	84	104	0.926	1.08	3.94
37. ZUMBRO RIVER	2.61	0.892	8	9	0.957	0.93	4.48
38. WHITEWATER RIVER	4.30	0.706	8	10	0.909	1.08	3.60
40. ROOT RIVER	3.34	0.947	13	13	0.991	0.49	5.23
41. SOUTH FORK ROOT RIVER	4.38	0.799	9	9	0.987	0.44	3.88
Average	0.90	1.057	35	44	0.924	1.51	5.58

Table 6. Results of yearly water and air temperature analysis.

Stream Station	A (°C)	B	N_c	N_{TOT}	R^2	S_p (°C)	S_e (°C)
3. PARTRIDGE RIVER	3.80	0.832	8	10	0.766		
8. RED RIVER OF THE NORTH	1.88	1.053	4	4	0.973	0.28	1.66
11. KAWISHIWI RIVER	–4.57	1.485	13	16	0.552	1.17	1.74
13. DUNKA RIVER	–1.14	1.173	5	6	0.951	0.24	1.07
19. MISSISSIPPI RIVER	0.14	1.045	9	12	0.428	1.08	1.43
20. MISSISSIPPI RIVER	–1.55	1.157	6	9	0.690	0.85	1.53
25. MINNESOTA RIVER	6.67	0.736	10	14	0.159	0.75	0.82
28. MISSISSIPPI RIVER	2.79	0.941	30	31	0.389	0.82	1.04
29. MISSISSIPPI RIVER	8.86	0.561	9	11	0.133	0.85	0.91
30. MISSISSIPPI RIVER	–5.11	1.356	9	10	0.833	0.66	1.62
31. MISSISSIPPI RIVER	–4.59	1.320	9	12	0.750	0.68	1.37
32. MISSISSIPPI RIVER	5.33	0.763	11	15	0.650	0.65	1.10
34. ST. CROIX RIVER	–5.74	1.411	4	6	0.918	0.21	0.72
35. MISSISSIPPI RIVER	–0.44	1.139	13	14	0.853	0.45	1.19
36. VERMILLION RIVER	3.07	0.739	14	15	0.688	0.66	1.17
Average	2.17	1.047	10	12	0.586	0.70	1.16

Table 7. Results of lumped data analysis.

Time Scale	A (°C)	B	N_c	N_{TOT}	R^2	S_p (°C)	S_e (°C)
DAILY	4.40	0.809	38,082	49,257	0.701	3.43	6.28
WEEKLY	1.88	0.966	5,465	7,112	0.830	2.58	6.25
MONTHLY	0.65	1.044	1,253	1,602	0.893	1.91	5.83
YEARLY	3.28	0.886	194	258	0.547	1.25	1.86

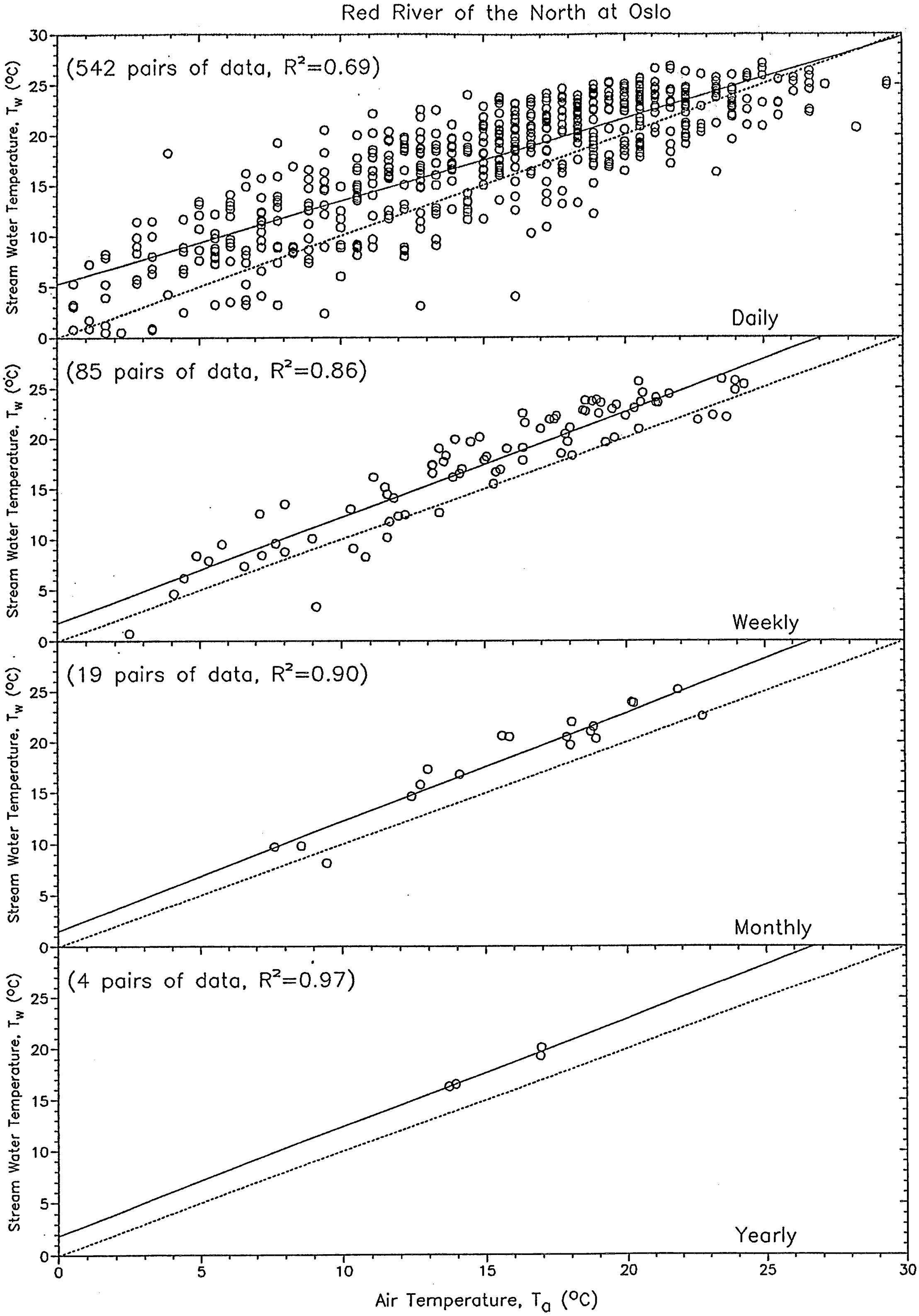

Fig. 1. Stream water temperature and air temperature correlation for the Red River of the North at Oslo, Minn. Mean daily, weekly, monthly, and yearly temperatures are plotted as circles. A solid line is the best fit line, and a dashed line equates water and air temperatures.

correlation on a daily scale, possibly because of a cooling water input. Station No. 43, also near Babbitt, Minn., gave an unreasonable constant on a weekly scale ($A = -10.9$). The data from these stations were therefore eliminated from consideration.

The regression lines for the remaining 39 individual streams are given in Figs. 2 to 5. The daily regression lines, shown in Fig. 2, are somewhat scattered about the one-to-one line. (Associated parameter values for the daily time scale are listed in Table 3.) For weekly and monthly time scales (Figs. 3 and 4; Tables 4 and 5), the regression lines are more consistent with one another than for the other time scales. Although some variation is expected, the correlations remained remarkably consistent across the entire state of Minnesota.

On a yearly time scale, the results were less satisfactory, and only 15 streams had sufficient data for correlation analysis. Fig. 5 shows the regression lines to be widely scattered. In contrast to the other time scales, a wide variety of slopes are observed, most likely because the number of data points was too small.

Lumped Data Set

The data points for all streams were lumped together to generate a general set of correlations. By lumping the data, the differences between individual streams were, of course, lost. A total of 38,082 data points were used. The data points for all four time scales are plotted in relation to the regression lines in Fig. 6.

On a daily time scale, the many data points are scattered rather far from the main regression line, but they follow the general slope enough for a correlation coefficient of 0.70. The weekly time scale plot is more consistent, with $R^2 = 0.83$. The monthly time scale produces a remarkably strong correlation: the slope of the regression line (1.04) is nearly 1, and the data is tightly packed along the entire regression line, resulting in a correlation coefficient of 0.89. On a yearly time scale, the data is concentrated in a smaller region and fewer data points (194) are available. The mean yearly temperatures are fairly consistent. As a result, deviations from the regression line have a more significant effect, resulting in a lower correlation coefficient of 0.55.

The lumped regression lines for the four time scales are shown superimposed on each other in Fig. 7. The water/air temperature relationships obtained from the lumped data analysis are as follows:

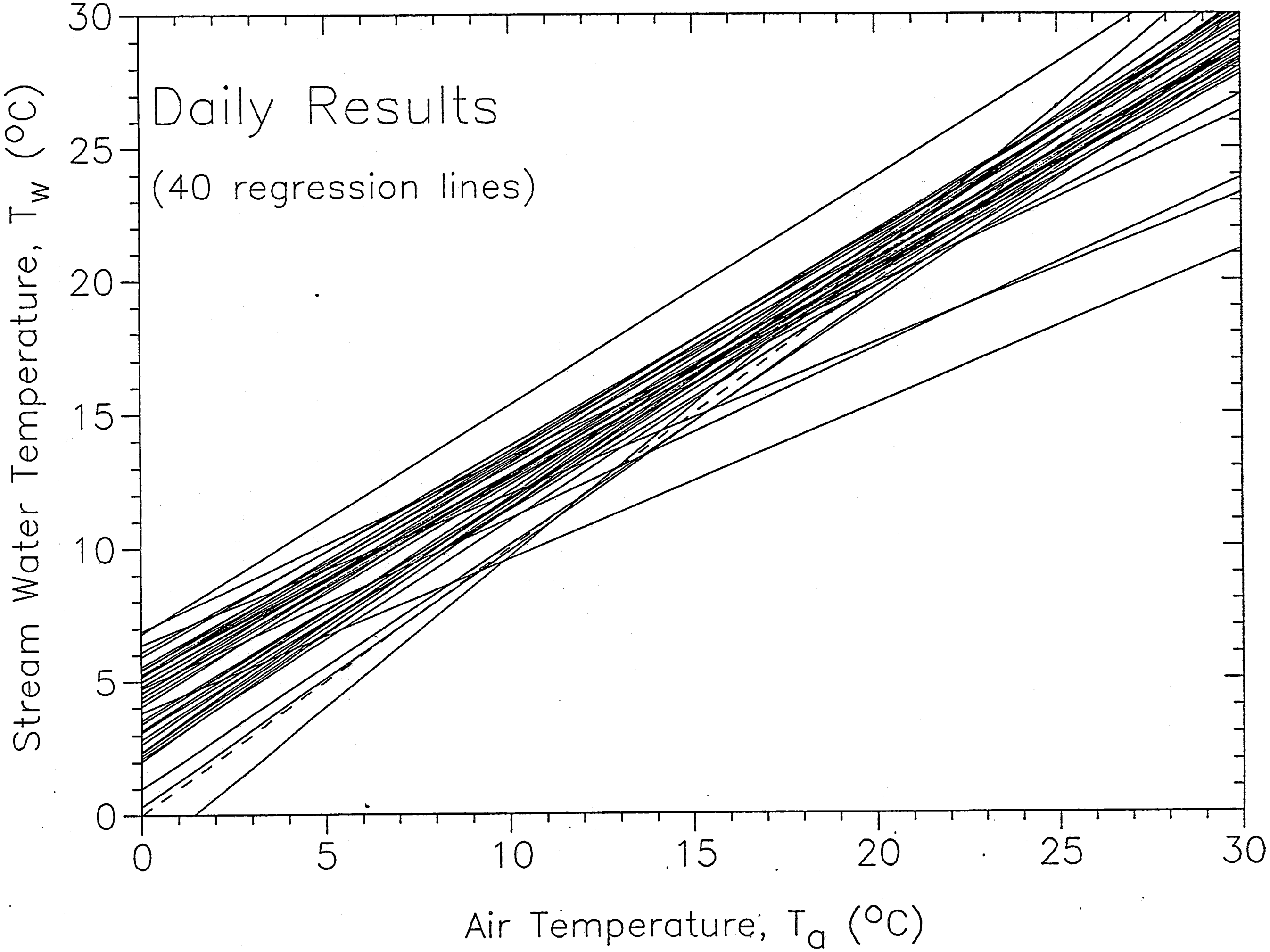

Fig. 2. Regression lines for the daily stream water temperatures and air temperatures. Forty regression lines are given for individual streams.

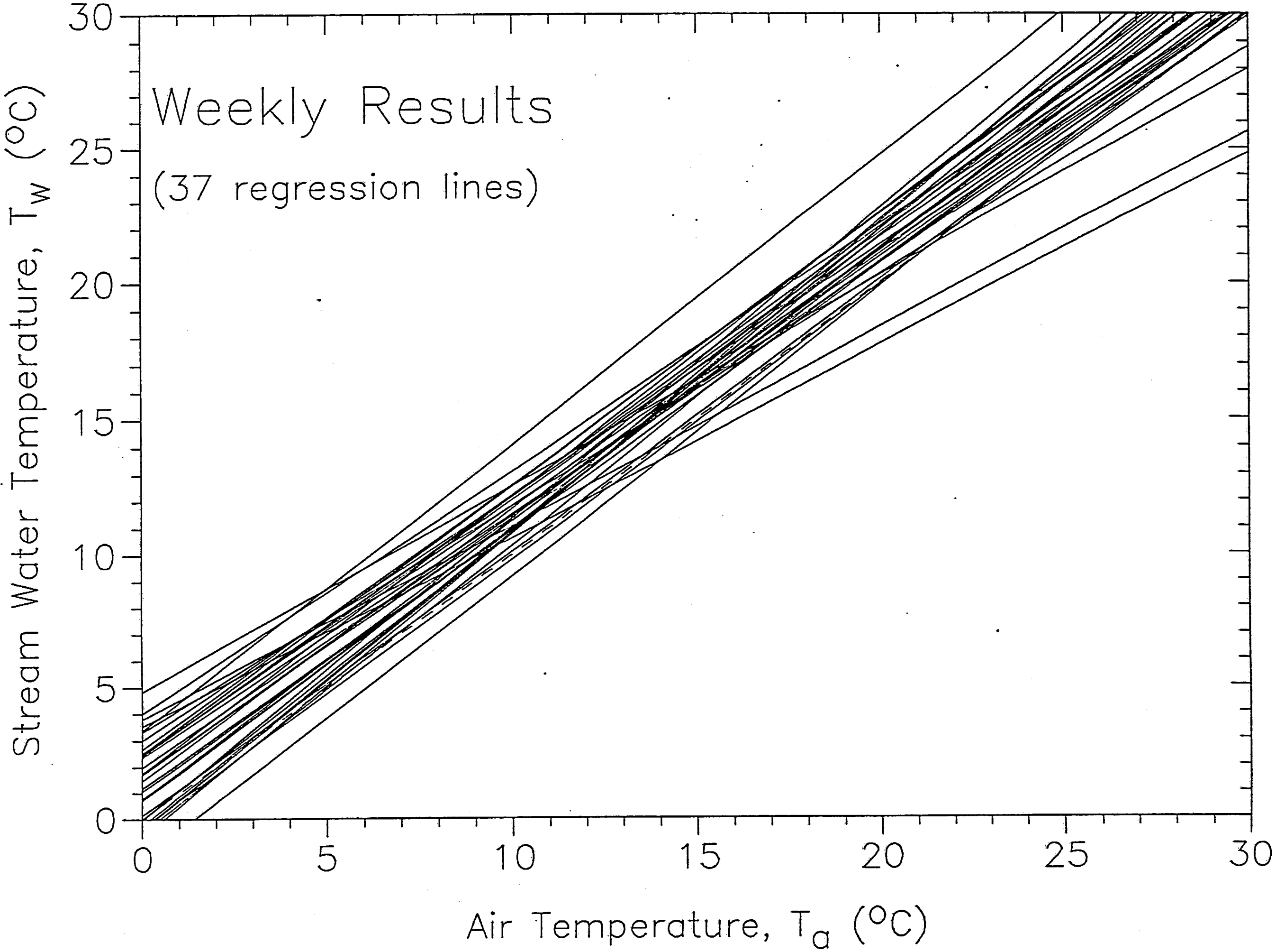

Fig. 3. Regression lines for the weekly stream water temperatures and air temperatures. Thirty-seven regression lines are given for individual streams.

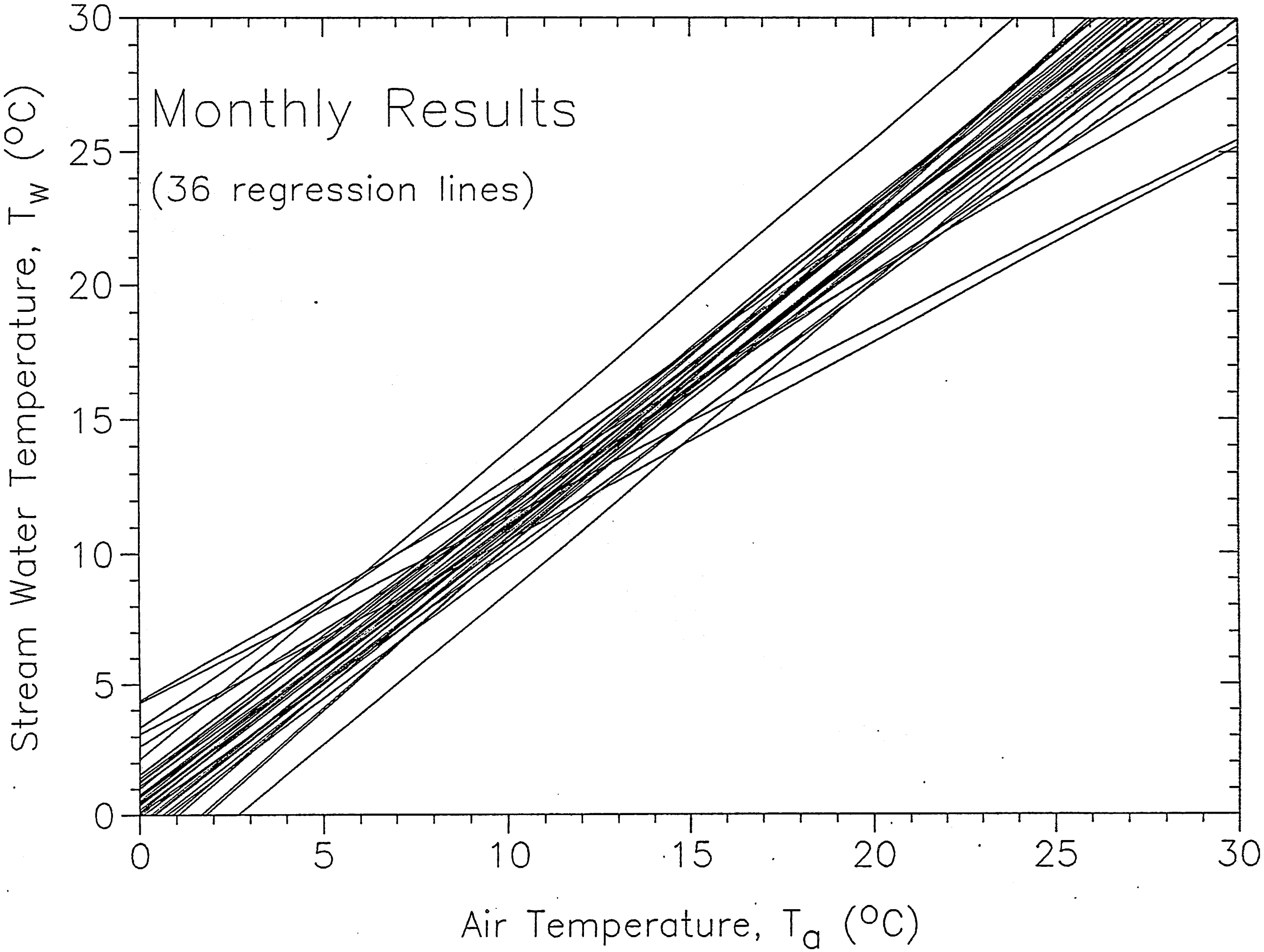

Fig. 4. Regression lines for the monthly stream water temperatures and air temperatures. Thirty-six regression lines are given for individual streams.

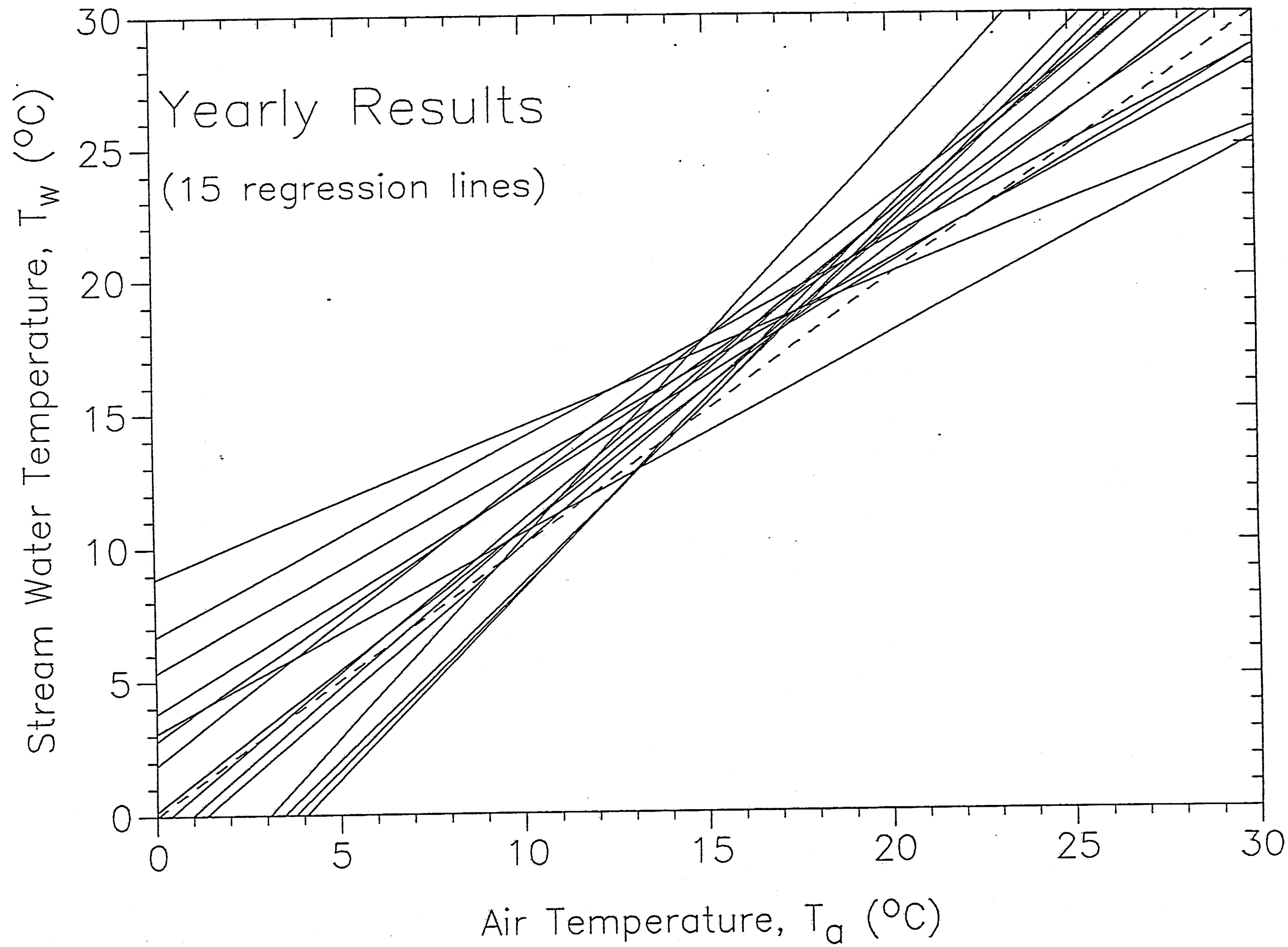

Fig. 5. Regression lines for the yearly stream water temperatures and air temperatures. Fifteen regression lines are given for individual streams.

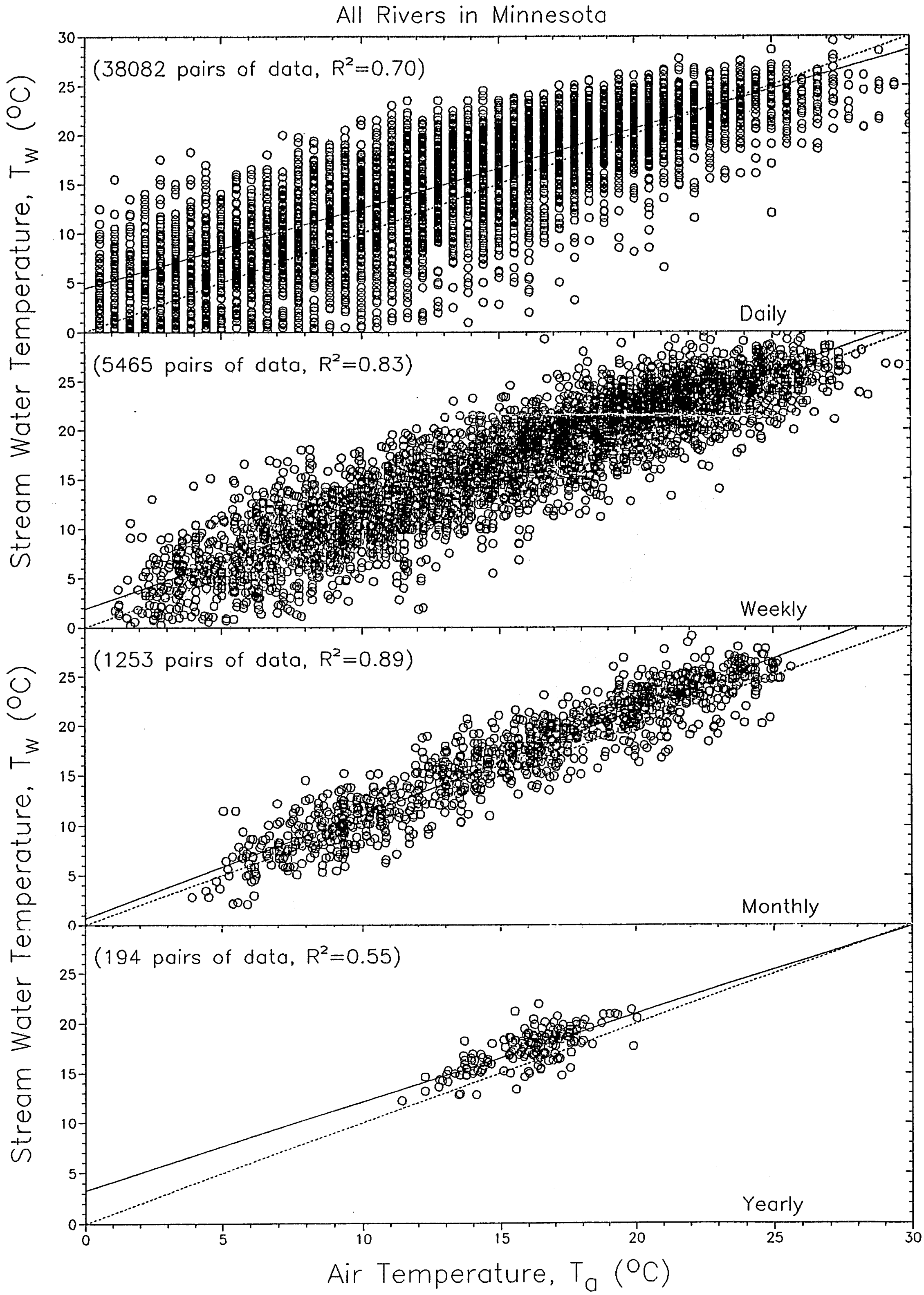

Fig. 6. Stream water and air temperature correlation for the lumped data set. Mean daily, weekly, monthly, and yearly temperatures are plotted as circles. A solid line is the best fit line, and a dashed line equates water and air temperature.

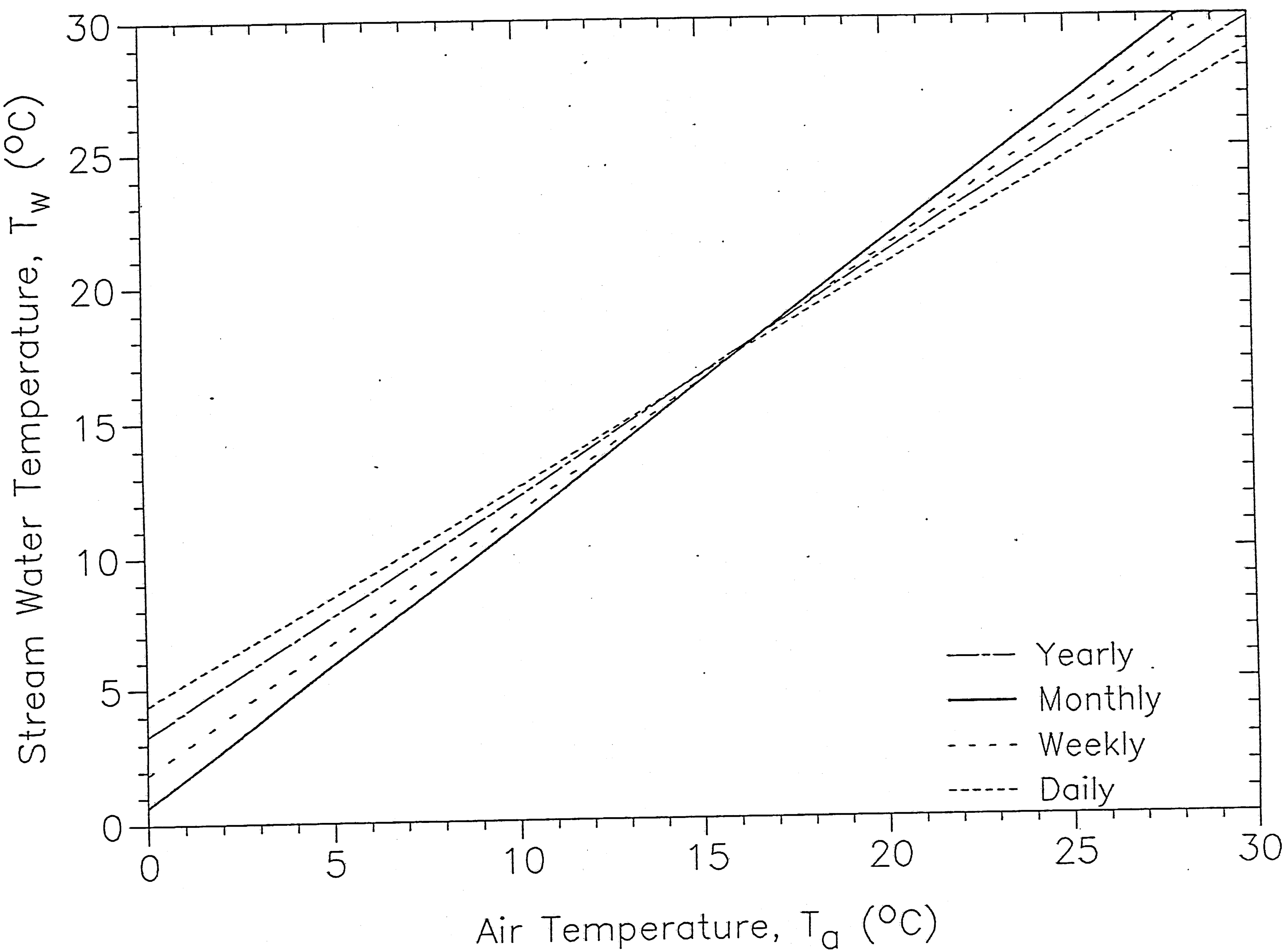

Fig. 7. Regression lines between stream water temperatures and air temperatures for the lumped data set. Correlations for mean daily, weekly, monthly, and yearly data are shown.

Daily:	$T_w = 4.4 + 0.81T_a$	with $S_p = 3.4°C$	(6a)
Weekly:	$T_w = 1.9 + 0.97T_a$	with $S_p = 2.6°C$	(6b)
Monthly:	$T_w = 0.7 + 1.04T_a$	with $S_p = 1.9°C$	(6c)
Yearly:	$T_w = 3.3 + 0.89T_a$	with $S_p = 1.3°C$	(6d)

The parameter values obtained from the lumped data analysis are listed in Table 7.

Analysis and Interpretation of Results

Analysis of lumped data

The assumption that a linear correlation exists between stream water temperature and air temperature is validated by the data plots in Fig. 6. It must be recognized, however, that a linear relationship with constant coefficients holds only for air temperatures ranging from slightly above 0°C to 30°C, the extent of the data. At air temperatures substantially below zero, ice cover forms and stream temperatures remain constant near 0°C.

The findings from the lumped data analysis generally agree with our analysis of the distribution of individual stream regressions, as we conclude that weekly and monthly time scales provide better correlations (i.e. higher R^2 and lower S_p values) than daily and yearly data. The lumped data appear to give the best correlations on a monthly time scale, where $R^2 = 0.89$. The monthly results are also the most consistent over a wide range of temperatures (4°C to 26°C in Fig. 6). The standard error of prediction of mean monthly stream temperatures is 1.9°C (Table 7), compared to a standard deviation of 5.8°C from the mean for the actual data. The standard error of prediction is only 1.3°C for the yearly time scale, but the temperature range is much narrower (11°C to 21°C) and the R^2 value is only 0.55.

It is also noteworthy that the four relationships for different time scales (Eqs. 6a to 6d) give the same water temperature (17.8°C) when the air temperature is approximately 16.3°C—roughly the mean annual air temperature. This is inherent in the linear regression analysis procedure.

As the time scale of the data lengthens from daily to weekly to monthly, it is remarkable that the constant A diminishes from 4.4 to 1.9 to 0.7°C and the slope B increases from 0.81 to 0.97 to 1.04, respectively. This compares to constants of $A = 5.1$ and $A = 2.9°C$ and slopes of $B = 0.75$ and $B = 0.86$ obtained by Stefan and Preud'homme (1993) for daily and weekly data from 11 central U.S. streams. Thus the trends are the same as in the Stefan and Preud'homme study, but the numerical values are somewhat different.

Analysis of individual streams

The relationship between air and water temperatures is expected to differ from stream to stream because of varying degrees of shading and differences in the source of the water (e.g. surface runoff, groundwater, reservoirs, etc.). For this reason, it is important to study each stream separately. Linear regressions for individual streams give better water temperature predictions than the lumped data correlations (Eqs. 6a to 6d). For individual streams, the standard errors of prediction S_p are, on average, about 0.4°C smaller than the S_p values for the lumped data set.

The results for individual streams on daily, weekly and monthly times scales are compared to other studies in Tables 8 to 10. Webb (1992) observed that "air/water temperature relationships become more scattered and less reliable as the time period over which the data are averaged becomes shorter." In a summary of least-squares linear regression relationships between mean air and water temperatures for 36 streams across the U.K., Webb gives the average values for A, B, and R^2 shown in Tables 8 to 10.

For the monthly time scale (Table 10), the results show great similarity between the streams in Minnesota and in the U.K. This is remarkable since the climates in those regions are quite different: Minnesota's is continental, whereas the U.K.'s is maritime. Mean monthly averages seem to hide these climate differences. On a weekly time scale the similarities are almost as strong, but differences become more pronounced on a daily time scale. For further comparison, the results obtained by Stefan and Preud'homme (1993) for 11 streams in the Mississippi River basin are also shown in Tables 8 and 9.

Table 8. Comparisons of linear regressions for daily water and air temperatures.

Streams/Location	*A* (°C) Avg.	Min.	Max.	*B* Avg.	Min.	Max.	R^2 (°C) Avg.	Min.	Max.
Minnesota (39 streams) [1]	4.2	–1.6	6.9	0.82	0.56	0.95	0.71	0.45	0.83
U.K. (36 streams) [2]	2.1	1.7	2.5	0.85	0.80	0.91	0.88	0.86	0.90
Mississippi River basin (11 streams) [3]	5.1	3.7	7.4	0.75	0.60	0.88	0.80	0.60	0.90

[1] Present study
[2] Webb (1987, 1992)
[3] Stefan and Preud'homme (1993)

Table 9. Comparisons of linear regressions for weekly water and air temperatures.

Streams/Location	*A* (°C) Avg.	Min.	Max.	*B* Avg.	Min.	Max.	R^2 (°C) Avg.	Min.	Max.
Minnesota (37 streams) [1]	1.7	–1.5	4.8	0.99	0.71	1.15	0.85	0.67	0.96
U.K. (36 streams) [2]	1.0	–0.3	3.7	1.00	0.65	1.16	0.91	0.86	0.97
Mississippi River basin (11 streams) [3]	2.9	0.4	5.4	0.86	0.67	1.03	0.89	0.75	0.97

[1] Present study
[2] Webb (1987, 1992)
[3] Stefan and Preud'homme (1993)

Table 10. Comparisons of linear regressions for monthly water and air temperatures.

Streams/Location	A (°C) Avg.	A (°C) Min.	A (°C) Max.	B Avg.	B Min.	B Max.	R^2 (°C) Avg.	R^2 (°C) Min.	R^2 (°C) Max.
Minnesota (36 streams) [1]	0.9	–3.1	4.4	1.06	0.71	1.23	0.92	0.84	0.99
U.K. (36 streams) [2]	2.0	–0.3	4.5	0.89	0.51	1.16	0.92	0.59	0.99
Mississippi River basin (11 streams) [3]	—	—	—	—	—	—	—	—	—
Krems River, Austria[4]	2.44			0.69			0.97		

[1] Present study

[2] Webb (1987, 1992)

[3] Stefan and Preud'homme (1993)

[4] Webb and Nobilis (1995)

Conclusions

The objective of this paper was to examine the accuracy of stream temperature estimates from simple linear regressions with air temperatures. The following is a summary of the findings.

Daily, weekly, monthly, and yearly stream water temperatures of 39 Minnesota streams could be estimated by linear regression with air temperature records. For the ensemble of 39 streams used in this study, standard deviations between measurements and predictions based on Eqs. 6a to 6d were on the order of 3.5°C (daily), 2.6°C (weekly), 1.9°C (monthly), and 1.3°C (yearly). Better results were obtained for weekly and monthly data than for daily data.

The dependence of regression coefficients on stream characteristics and weather parameters other than air temperature is evident in the results but difficult to quantify without resorting to more complex heat budget models. Complete heat budget models can predict stream temperatures with standard deviations of 1°C or less (Stefan et al., 1980; Sinokrot and Stefan, 1992). More input data would be required for this purpose. The relationships found in this study can be used to obtain rough estimates of stream water temperatures.

One of the greatest drawbacks of using generic linear regression models to predict stream water temperature from air temperature is the lack of damping by thermal inertia or shading effects. If regression equations can be developed from data for individual streams, these effects are included in the coefficient values, and the standard errors of prediction are usually reduced. The average reduction was found to be from 0.2°C to 0.6°C depending on timescale.

References

Brown, G.W., 1971. An Improved Temperature Prediction Model for Small Streams. Completion Report, Oregon Water Resources Research Institute, Corvallis, Oregon, 20 pp.

Crisp, D.T. and G. Howson, 1982. Effect of Air Temperature upon Mean Water Temperature in Streams in the North Pennines and English Lake District. *Freshwater Biology* 12:359–367.

Jobson, H.E., 1973. The Dissipation of Excess Heat from Water Systems. ASCE, *Journal of the Power Division,* 99(PO1):89–103.

Morin, G., D. Couillard, D. Cluis, H.G. Jones, and J.-M. Gauthier, 1987. Prediction of Temperatures in Rivers using a Conceptual Model (previson des temperatures de l'eau en riviere a l'aide d'un modele conceptuel). *Hydrological Sciences Journal* HSJODN 23(1):31–41.

Noble, R.D., 1979. Analytical Prediction of Natural Temperatures in Rivers. Technical Note, ASCE, *Journal of the Environmental Engineering Division* 105(EE5):1014–1018.

Novotny, V. and P.A. Krenkel, 1973. Simplified Mathematical Model of Temperature Changes in Rivers. *Journal of Water Pollution Control Federation* 45(2):239–248.

Paily, P.P. and E.O. Macagno, 1976. Numerical Predictions of Thermal Regime of Rivers. Proceedings Paper 11997, ASCE, *Journal of the Hydraulics Division* 102(HY3).

Preud'homme, E.B. and H.G. Stefan, 1992. Relationship Between Water Temperatures and Air Temperatures for Central U.S. Streams. Project Report No. 333, St. Anthony Falls Laboratory, University of Minnesota, Minneapolis, Minnesota, 141 pp.

Sinokrot, B.A. and H.G. Stefan, 1992. Stream Temperature Dynamics: Measurements and Modeling. *Water Resources Research* 29(7):2299–2313.

Sinokrot, B.A., H.G. Stefan, J.H. McCormick, and J.G. Eaton, 1995. Modeling of Climate Change Effects on Stream Temperatures and Fish Habitat Below Dams and Near Groundwater Inputs. *Climatic Change* 30:181–200.

Smith, K., 1981. The Prediction of River Water Temperature. *Hydrological Science Bulletin* 26(1):19–32.

Stefan, H.G., J. Gulliver, M.G. Hahn, and A.Y. Fu, 1980. Water Temperature Dynamics in Experimental Field Channels: Analysis and Modeling. Project Report No. 193, St. Anthony Falls Laboratory, University of Minnesota, Minneapolis, Minnesota.

Stefan, H.G. and E.B. Preud'homme, 1993. Stream Temperature Estimation from Air Temperature. *Water Resources Bulletin* 29(1): 27–45.

Webb, B.W., 1987. The Relationship between Air and Water Temperatures for a Devon River. Rep. Trans., Devon Association for the Advancement of Science, 119:197–222.

Webb, B.W., 1992. Climate Change and the Thermal Regime of Rivers. Report for the Department of the Environment under contract PECD/7/7/348, University of Exeter, U.K., Department of Geography, 79 pp.

Webb, B.W. and F. Nobilis, 1995. A long-term perspective on the nature of the air-water temperature relationship, in Journal of Hydrological Processes, D. E. Walling (ed.)

Appendix - Computer Programs

The computer programs first dealt with several issues with regard to the format of the temperature data before the correlation stage could be reached. The first such issue had to do with the stream water temperature records which, as received from the EPA, were stored in a single, compact file. This file combined the data for all streams, one after another, with temperature readings for eight consecutive days on a line. A further complication was that for each stream, the lines of the file were not purely chronological; a series of lines containing maximum temperature readings for each day would be followed by another series of lines with minimum temperatures for the same days, and in some cases mean temperatures for the same time period would follow. The program `SPLIT.FOR` was used to divide this large file into a series of individual files for each of the 43 streams. Each file would then contain a single temperature reading on a line, along with the type of data represented (maximum, minimum, mean, and others) and the date of the reading. The problem of sorting the data chronologically was addressed by the next program in the sequence.

After separating the large stream temperature data file into individual files, data for each stream were linked to air temperature data from the nearest monitoring station. This was done in a two-step process using the program `SETUP.FOR`. First, each line of a file for a stream was read in. The temperature reading was then stored in an array which was selected based on whether the reading was a maximum, minimum, or mean temperature for the day. At the same time, the selected air temperature data file was searched until readings—maximum, minimum, and mean, all on the same line—for that same date were found. After reading in the entire stream temperature file, the temperature data was written out to another file in chronological order, with all air and water maximum, minimum, and mean temperatures for a day on a single line.

The second step handled by `SETUP.FOR` was to generate mean temperatures for each of the time scales: daily, weekly, monthly, and yearly. Here, the data set from the last step was reduced to the period between April 1 and October 31. Daily mean temperatures were extracted if available, or calculated by averaging the maximum and minimum temperature readings for a day. To find the weekly mean temperatures, the daily mean temperatures were averaged over the course of each week, beginning with the week of April 1. A similar procedure was followed for each month and each year, from April to October inclusive, to generate monthly and yearly mean temperatures. In the process, mean temperature data was accepted only if daily temperatures were present for the majority of the days in the time period (4 days per week, 16 days per month, or 107 days per year between April and October). Data that did not fulfill this criterion were ignored along with a few odd readings that fell below 0°C, where ice cover might still have been present. The result was four separate data files for each stream, each containing the mean air and water temperatures for a given time scale.

The actual correlation between stream water and air temperature was carried out using the program `CORREL.FOR`. The temperatures for each stream and time scale were read in and the constants, A and B in Eq. 1, were calculated using least-squares regression. This correlation was only performed if there were at least 4 data points available for a given stream and time scale. Then, the file was read in a second time so that the measured stream temperatures could be compared to the temperatures predicted by the model. From this, the correlation coefficient R^2, standard error of prediction S_p, and standard deviation of the mean S_e were calculated.

Finally, the program performed a lumped correlation using the data from of all the streams simultaneously. Again, the constants A and B were calculated, and each file was then read in once more to determine the R^2 values for this lumped correlation. Once this was completed, the results were written out to a single file, divided into five sections, containing correlations for each stream on daily, weekly, monthly, and yearly time scales along with the lumped correlation for the same time scales.

```
C ***********************************************************************
      PROGRAM SPLIT
C ***********************************************************************
C Author: John Pilgrim
C Date:   3/21/95

C Purpose
C The program reads in a data file, provided by the EPA in Duluth,
C containing stream temperature data at a number of stations throughout
C Minnesota.
C The data file contains readings for 8 consecutive days on a single
C line. This program splits the file in a number of separate files for
C each stream while writing out the data in a format used by the next
C program in the sequence, SETUP, with one temperature reading per line

C Description of variables
C STATNO(i):       Array of station ID numbers, given by input file
C STNAME(i):       Array of station names
C LAT(i), LONG(i): Arrays of latitudinal, longitudinal coordinates

C Subroutines called
C READST: Reads in station information from file MNSTN.TXT
C EXTRAC: Extracts temperature data from file MNTMP.TXT into a series o:
C         individual files

C Variable declarations
      CHARACTER STATNO(43) *8, STNAME(43) *48
      REAL LAT(43), LONG(43)

C Read in station information, get relevant parameters

      CALL READST (STATNO, STNAME, LAT, LONG)

C Extract temperature data; pass on station information so each output
C file can be labeled clearly

      CALL EXTRAC (STATNO, STNAME, LAT, LONG)

      END

C ***********************************************************************
      SUBROUTINE READST (STATNO, STNAME, LAT, LONG)
C ***********************************************************************

C Purpose
C Reads in station information from data file MNSTN.TXT, provided by th
C EPA in Duluth. The essential parameters describing each station are
C passed back to the main program.
C Please note that all double quotes (") in the file MNSTN.TXT must first
C be changed to single quotes (') in order for this program to make use
C of the file.

C Description of variables
C Output
C STATNO(i):       Array of station ID numbers, given by input file
C STNAME(i):       Array of station names
C LAT(i), LONG(i): Arrays of latitudinal, longitudinal coordinates
C
C Local
C I:  Counter variable

C Declarations for calling arguments
      CHARACTER STATNO(43) *8, STNAME(43) *48
      REAL LAT(43), LONG(43)

C Local declarations
      INTEGER I

C Read in information from input file MNSTN.TXT

      OPEN (UNIT=1, FILE='MNSTN.TXT')

      DO 10 I = 1, 43
         READ (1,*) STATNO(I), STNAME(I), LAT(I), LONG(I
   10 CONTINUE

      CLOSE (UNIT=1)

      RETURN
```

```
      END

C ***********************************************************************
      SUBROUTINE EXTRAC (STATNO, STNAME, LAT, LONG)
C ***********************************************************************

C Purpose
C Extracts temperature data from data file MNTMP.TXT into a series o:
C individual files, one for each station. The output files are labeled
C 'MNTMP*.DAT,' where * represents a number from 1 to 43. Relevant
C information to identify each station, read in by READST, is writte:
C out as a header for each file. Excess data, from stations which have
C no information in MNSTN.TXT, is written out to MNTMP99.DAT.
C While the input file contains readings for 8 consecutive days on a
C single line, the temperature data is written out with one reading per
C line.

C Description of variables
C Input
C STATNO(i):       Array of station ID numbers, given by input file
C STNAME(i):       Array of station names
C LAT(i), LONG(i): Arrays of latitudinal, longitudinal coordinates
C
C Local
C FNAME(i):  Array of output filenames
C NUM:       Two-digit file identifier (01-43), represented as character
C STNID:     Station ID at which a particular set of readings were taken
C STAT:      Statistical code representing the type of data:
C                00001:   Maximum value for each day
C                00002:   Minimum value for each day
C                00003:   Mean value for each day
C                00004:   AM (Reading obtained during morning hours)
C                00005:   PM (Reading obtained during evening hours)
C F:         Marks records with gross errors or unacceptable values
C YEAR:      Calendar year date
C MO:        Month designation (1-12)
C SE:        Segment of month divided into four parts. With readings
C            for 8 days on a line, and 4 lines/segments per month, this
C            allows for 32 possible readings.
C DAY:       Actual day of month, calculated from SE
C VAL(i):    Temperature readings in degrees C
C STN:       Station/file identifier as integer
C FID:       File ID unit number (1-3)
C I:         Counter variable

C Declarations for calling arguments
      CHARACTER STATNO(43) *8, STNAME(43) *48
      REAL LAT(43), LONG(43)

C Local declarations
      CHARACTER FNAME(43) *12, NUM *2
      CHARACTER STNID *8, STAT *5, F *
      INTEGER YEAR, MO, SE, DAY
      REAL VAL(8)
      INTEGER STN, FID, I

C Assign names of output files

      DO 10 I = 1, 43
         NUM(1:1)  = CHAR(MOD(I/10, 10) + 48)
         NUM(2:2)  = CHAR(MOD(I,10) + 48)
         FNAME(I)  = 'MNTMP' // NUM // '.DAT'
   10 CONTINUE

C Open input and output files, write header to MNTMP01.DAT
C (Assume STATNO(1) is the first station found in input file)

      OPEN (UNIT=1, FILE='MNTMP.TXT')
      OPEN (UNIT=2, FILE='SOURCES\MNTMP99.DAT')
      OPEN (UNIT=3, FILE= ('SOURCES\'//FNAME(1)) )

      STN = 1

      WRITE (*,*) FNAME(STN)
      WRITE (3,100) STATNO(STN), STNAME(STN), LAT(STN), LONG(STN)

C Loop for reading from input file, extracting to separate output files

      DO WHILE (.TRUE.)
```

```
          READ (1,110, END=99) STNID, STAT, YEAR, MO, SE, VAL(1),
     +    VAL(2), VAL(3), VAL(4), VAL(5), VAL(6), VAL(7), VAL(8), F

C Make sure the data just read is from the station being written out
C (STNID should match station ID of open output file)

          IF (STNID .NE. STATNO(STN)) THEN

C New station found in input file. Close the current output file; searcl
C for correct output file, setting file identifier to 0 until the match
C is found.

             CLOSE (UNIT=3)
             STN = 0

             DO 20 I = 1, 43
                IF (STNID .EQ. STATNO(I)) THEN

C Matching station ID found. Write header and direct output to this fil

                   STN = I
                   OPEN (UNIT=3, FILE= ('SOURCES\'//FNAME(STN)) )
                   WRITE (*,*) FNAME(STN)
                   WRITE (3,100) STATNO(STN), STNAME(STN)
     +                          LAT(STN), LONG(STN)
                ENDIF
   20        CONTINUE

          ENDIF

C If no match found after this search, set FID for output to MNTMP99.DA
C (unit #2); otherwise write to proper output file (unit #3)

          IF (STN .EQ. 0) THEN
             FID = 2
          ELSE
             FID = 3
          ENDIF

C A double quote (") is read into F if there is no error code; this is
C cleared so that it does not appear in the output.

          IF (F .EQ. '"') THEN
             F = ' '
          ENDIF

C Write out temperature readings. The true day of the month is
C calculated, and readings are written out on one line for each day.

          DO 30 I = 1, 8
             IF (VAL(I) .NE. 9999.0) THEN
                DAY = 8*(SE-1) + I
                WRITE (FID,120) STNID,STAT, MO,DAY,YEAR-1900, VAL(I), F
             ENDIF
   30     CONTINUE

C End of reading/extraction loop (or end of input file encountered).
C Close all open files and exit subroutine.

       ENDDO

   99  CONTINUE

       CLOSE (UNIT=1)
       CLOSE (UNIT=2)
       CLOSE (UNIT=3)

  100  FORMAT (A8,',',A48,',',F8.5,',',F8.5)
  110  FORMAT (1X, A8, 3X, A5, 3X, I4, 3X, I2, 3X, I2, 3X
     +         F7.2, 3X, F7.2, 3X, F7.2, 3X, F7.2, 3X,
     +         F7.2, 3X, F7.2, 3X, F7.2, 3X, F7.2, 3X, A1
  120  FORMAT (A8,',',A5,',',I2.2,'/',I2.2,'/',I2,',',F7.2,',',A1)

       RETURN
       END
```

```
C ************************************************************************
     PROGRAM SETUP
C ************************************************************************
C Author: John Pilgrim
C Date:   5/9/95

C Purpose
C The program reads in air temperature and stream temperature data files
C for a number of stations throughout Minnesota. Air temperatures are
C taken from a series of files provided by the Midwest Hydrology Center
C Stream temperatures come from the data file created by the previous
C program in the sequence, SPLIT.
C For each stream, four output files are created, listing the mean air
C and stream temperatures side by side for each time scale: daily,
C weekly, monthly, and yearly. These files are then used by the final
C program in the sequence, CORREL, to determine the correlation
C coefficients.

C Description of variables
C AIRFN(i): Array of air temperature filenames
C WATFN(i): Array of stream temperature filenames
C MIDFN(i): Array of intermediate filenames (output)
C OUTFN(i): Array of mean temperature output filenames
C TYPE:     Time scale for correlation
C           (1: daily; 2: weekly; 3: monthly; 4: yearly)
C I:        Counter variable

C Subroutines called
C ASSIGN: Assigns filenames to the arrays
C COMBIN: Combines air and stream temperatures from input files;
C         generates intermediate file containing only the days of
C         interest, to be used in RELATE subroutine
C RELATE: Determines mean temperatures on each time scale; creates
C         output files with this information

C Variable declarations
      CHARACTER AIRFN(43) *10, WATFN(43) *12
      CHARACTER MIDFN(43) *12, OUTFN(43,4) *12
      INTEGER I, TYPE

C Assign filenames to arrays. Note that the air temperature data chosen
C should give the best correlations with the streams.

      DATA AIRFN /
     + '213282.OUT','218419.OUT','210390.OUT','210390.OUT','211630.OUT',
     + '211891.OUT','211891.OUT','210252.OUT','213455.OUT','213455.OUT',
     + '219101.OUT','210390.OUT','210390.OUT','210390.OUT','210390.OUT',
     + '215325.OUT','215435.OUT','215435.OUT','215435.OUT','217377.OUT',
     + '215400.OUT','211263.OUT','214176.OUT','214176.OUT','214176.OUT',
     + '215435.OUT','217377.OUT','217377.OUT','217377.OUT','217377.OUT',
     + '217377.OUT','217377.OUT','217377.OUT','217377.OUT','217377.OUT',
     + '212737.OUT','219067.OUT','219067.OUT','219067.OUT','219067.OUT',
     + '219067.OUT','210390.OUT','210390.OUT' /

      CALL ASSIGN (WATFN, MIDFN, OUTFN)

      WRITE (*,*) ' '

C Combine air and stream temperature data; make intermediate file,
C eliminating days with missing data or from winter months

      DO 10 I = 1, 43
         WRITE (*,*) MIDFN(I)
         CALL COMBIN (AIRFN(I), WATFN(I), MIDFN(I))
   10 CONTINUE

      WRITE (*,*) ' '

C Create output files with mean temperatures on different time scales

      DO 20 I = 1, 43
         DO 30 TYPE = 1, 4
            WRITE (*,*) OUTFN(I,TYPE)
            CALL RELATE (MIDFN(I), OUTFN(I,TYPE), TYPE)
   30    CONTINUE
   20 CONTINUE

      END
```

```
C ************************************************************************
      SUBROUTINE ASSIGN (WATFN, MIDFN, OUTFN)
C ************************************************************************

C Purpose
C Assigns names of data files into arrays. This is done for the input
C air temperature file, the intermediate file, and the final output
C files.

C Description of variables
C Output
C WATFN(i): Array of stream temperature filenames
C MIDFN(i): Array of intermediate filenames (output)
C OUTFN(i): Array of mean temperature output filenames
C
C Local
C NUM: Stream index represented as character
C I:   Counter variable

C Declarations for calling arguments
      CHARACTER WATFN(43) *12, MIDFN(43) *12, OUTFN(43,4) *12

C Local declarations
      CHARACTER NUM *2
      INTEGER I

C Assign filenames to arrays. Convert the 2-digit number I to character

      DO 10 I = 1, 43
         NUM(1:1) = CHAR(MOD(I/10, 10) + 48)
         NUM(2:2) = CHAR(MOD(I, 10) + 48)

C Concatenate strings to create filenames

         WATFN(I)   = 'MNTMP'  // NUM // '.DAT'
         MIDFN(I)   = 'STREAM' // NUM // '.DAT'
         OUTFN(I,1) = 'COR' // NUM // '_DY.DAT'
         OUTFN(I,2) = 'COR' // NUM // '_WK.DAT'
         OUTFN(I,3) = 'COR' // NUM // '_MO.DAT'
         OUTFN(I,4) = 'COR' // NUM // '_YR.DAT'
   10 CONTINUE

      RETURN
      END
```

```
C ************************************************************************
      SUBROUTINE COMBIN (AIRFN, WATFN, MIDFN)
C ************************************************************************

C Purpose
C Reads air and stream temperatures from input files and combines the
C data concisely in an intermediate file. All data is sorted by date.
C and days with missing data are ignored so only the important dates
C appear in the output. The file shows maximum, minimum, and mean air
C and stream temperatures side by side

C Description of variables
C Input
C AIRFN(i): Array of air temperature filenames
C WATFN(i): Array of stream temperature filenames
C MIDFN(i): Array of intermediate filenames
C
C Local
C STNAME:        Original 48-character string read in for air station
C SSTN:          Stream temperature station name
C ASTN:          True air temperature station name
C RECORD:        Dummy variable for reading from files
C TEMP:          Temperature value read from stream temperature file
C AMO, ADAY, AYR:      Date read from air temperature file
C TMAX, TMIN, TMEAN: Maximum, minimum, and mean air temperatures read
C SMO, SDAY, SYR:      Date read from stream temperature file
C DTYPE:               Stream temperature data type
C                      (1: maximum; 2: minimum; 3: mean
C ADATE:         Numerical value representing date from air file
C SDATE:         Numerical value representing date from stream file
C PREV:          Previous stream date; used in continuous check
C I:             Counter variable
```

```
C FLAG(m,d,y): Indicates whether stream data exists for a given date
C AMAX(m,d,y), AMIN(m,d,y), AMEAN(m,d,y): Air data for given date
C SMAX(m,d,y), SMIN(m,d,y), SMEAN(m,d,y): Stream data for given date

C Subroutines called
C PARSE:  Parses air station name from a 48-character string read in
C MIDOUT: Controls all output to intermediate file

C Declarations for calling arguments
      CHARACTER AIRFN *10, WATFN *12, MIDFN *12

C Local declarations
      CHARACTER STNAME *48, SSTN *48, ASTN *48, RECORD *1, TEMP *6
      INTEGER AMO, ADAY, AYR, TMAX, TMIN, TMEAN, SMO, SDAY, SYR, DTYPE
      INTEGER *4 ADATE, SDATE, PREV
      INTEGER I, FLAG(12,31,36)
      REAL AMAX(12,31,36), AMIN(12,31,36), AMEAN(12,31,36)
      CHARACTER *6 SMAX(12,31,36), SMIN(12,31,36), SMEAN(12,31,36)

C Open input and output files. Read in station names; parse air station
C name from the string read in

      OPEN (UNIT=1, FILE=('SOURCES\'//AIRFN))
      OPEN (UNIT=2, FILE=('SOURCES\'//WATFN))
      OPEN (UNIT=3, FILE=('STREAMS\'//MIDFN))

      READ (1,100) RECORD
      READ (1,110) STNAME
      READ (2,110) SSTN
      DO 10 I = 1, 4
         READ (1,100) RECORD
   10 CONTINUE

      CALL PARSE (STNAME, ASTN)

C Read in stream temperature from first day, to begin read loop

      READ (2,120) DTYPE, SMO, SDAY, SYR, TEMP
      SDATE = SYR*10000 + SMO*100 + SDAY

C Loop to read temperature data; linking it with the appropriate days

      DO WHILE (.TRUE.)
         PREV = SDATE

         DO WHILE (SDATE .EQ. PREV)

C Search for air data to link with stream data just read
C Initially, stream variables are cleared. As characters, they can be
C written out as blank spaces instead of 0's if values have not been set.

            IF (FLAG(SMO,SDAY,SYR-55) .EQ. 0) THEN
               SMAX(SMO,SDAY,SYR-55)  = '      '
               SMIN(SMO,SDAY,SYR-55)  = '      '
               SMEAN(SMO,SDAY,SYR-55) = '      '

C Search for an air date which matches the stream date. If found, set
C FLAG accordingly.

               DO WHILE (ADATE .NE. SDATE
                  READ (1,130, END=15) AYR, AMO, ADAY, TMAX, TMIN, TMEA
                  ADATE = AYR*10000 + AMO*100 + ADAY
               ENDDO
               FLAG(SMO,SDAY,SYR-55) = 1

C It is possible to read through to the end without finding the matchin
C air date, if it came earlier in the file. If this is the case, rewind
C the air file and search again.

   15          IF (FLAG(SMO,SDAY,SYR-55) .EQ. 0) THEN
                  REWIND (UNIT=1)
                  DO 20 J = 1, 6
                     READ (1,100) RECORD
   20             CONTINUE

                  DO WHILE (ADATE .NE. SDATE)
                     READ (1,130) AYR, AMO, ADAY, TMAX, TMIN, TMEAN
                     ADATE = AYR*10000 + AMO*100 + ADAY
                  ENDDO
                  FLAG(SMO,SDAY,SYR-55) = 1
               ENDIF

C If -99 appears for the mean air temperature, data is missing for that
C day. Set FLAG so that this day will be ignored in all other loops; or,
C convert air temperatures to degrees Celsius.

               IF (TMEAN .EQ. -99) THE
                  FLAG(SMO,SDAY,SYR-55) = -1
               ELSE
                  AMAX(SMO,SDAY,SYR-55)  = (REAL(TMAX) - 32.0) * 5.0/9.0
                  AMIN(SMO,SDAY,SYR-55)  = (REAL(TMIN) - 32.0) * 5.0/9.0
                  AMEAN(SMO,SDAY,SYR-55) = (REAL(TMEAN)- 32.0) * 5.0/9.0
               ENDIF
            ENDIF

C Set appropriate stream temperature variable based on type of data

            IF (DTYPE .EQ. 1) THEN
               SMAX(SMO,SDAY,SYR-55)  = TEMP
            ELSEIF (DTYPE .EQ. 2) THEN
               SMIN(SMO,SDAY,SYR-55)  = TEMP
            ELSEIF (DTYPE .EQ. 3) THEN
               SMEAN(SMO,SDAY,SYR-55) = TEMP
            ENDIF

C Read next stream data available; continue read loop

            READ (2,120, END=99) DTYPE, SMO, SDAY, SYR, TEMP
            SDATE = SYR*10000 + SMO*100 + SDAY

         ENDDO
      ENDDO

   99 CONTINUE

C End of stream file reached. Write out data to intermediate file;
C close all open files and exit subroutine.

      CALL MIDOUT (SSTN, ASTN, FLAG, AMAX, AMIN, AMEAN,
     +             SMAX, SMIN, SMEAN)

      CLOSE (UNIT=1)
      CLOSE (UNIT=2)
      CLOSE (UNIT=3)

  100 FORMAT (A1)
  110 FORMAT (9X, A48)
  120 FORMAT (9X, I5, 1X, I2, 1X, I2, 1X, I2, 2X, A6)
  130 FORMAT (2X, I2, 1X, I2, 1X, I2, 12X, I3, 5X, I3, 5X, I3)

      RETURN
      END


C ***********************************************************************
      SUBROUTINE PARSE (STNAME, ASTN)
C ***********************************************************************

C Purpose
C Parses the air station name from the 48-character string read in,
C which likely contains unwanted information following the name. The
C string returned contained only the station name, padded with blanks.

C Description of variables
C Input
C STNAME: Original 48-character string read in for air station
C
C Output
C ASTN:   True air station name to be returned
C
C Local
C I, J:   Counter variables (J picks up where I leaves off)

C Declarations for calling arguments
      CHARACTER *48 STNAME, ASTN

C Local declarations
      INTEGER I, J
```

```
C Assigns characters to the ASTN string. The station name in STNAME is
C one word, with underscores in place of spaces, so the first white
C space character found indicates the end of the station name.
C Underscores in the station names are replaced with true spaces.

      I = 1
      DO WHILE ((I .LE. 48) .AND. (STNAME(I:I) .NE. ' '))
         IF (STNAME(I:I) .EQ. '_') THEN
            ASTN(I:I) = ' '
         ELSE
            ASTN(I:I) = STNAME(I:I)
         ENDIF
         I = I + 1
      ENDDO

C ASTN now contains the complete air station name. Pad the rest of the
C string with blanks.

      DO 10 J = I, 48
         ASTN(J:J) = ' '
   10 CONTINUE

      RETURN
      END


C ***********************************************************************
      SUBROUTINE MIDOUT (SSTN, ASTN, FLAG, AMAX, AMIN, AMEAN,
     +                   SMAX, SMIN, SMEAN)
C ***********************************************************************

C Purpose
C Outputs maximum, minimum, and mean air and stream temperature data to
C intermediate files for each stream. All data is sorted by date, and
C days with missing data are ignored completely.

C Description of variables
C Input
C SSTN:         Stream temperature station name
C ASTN:         Air temperature station name
C FLAG(m,d,y): Indicates whether stream data exists for a given date
C AMAX(m,d,y), AMIN(m,d,y), AMEAN(m,d,y): Air data for given date
C SMAX(m,d,y), SMIN(m,d,y), SMEAN(m,d,y): Stream data for given date
C
C Local
C JDAY:    Julian day; ranges from 1 (January 1) to 365 (December 31)
C M, D, Y: Counter variables representing month, day, year

C Subroutines called
C JULIAN:  Determines the Julian day for a given day out of the year

C Declarations for calling arguments
      CHARACTER SSTN *48, ASTN *48
      INTEGER FLAG(12,31,36)
      REAL AMAX(12,31,36), AMIN(12,31,36), AMEAN(12,31,36)
      CHARACTER *6 SMAX(12,31,36), SMIN(12,31,36), SMEAN(12,31,36)

C Local declarations
      INTEGER *2 JDAY
      INTEGER M, D, Y

C Create file header

      WRITE (3,100) 'Stream data:  ', SSTN
      WRITE (3,100) 'Air temp data:', ASTN
      WRITE (3,110) ' '
      WRITE (3,120) 'Air temperature', 'Stream temperature'
      WRITE (3,130) 'Date     Julian', 'Tmax ', 'Tmin ', 'Tmean',
     +              'Tmax ', 'Tmin ', 'Tmean'
      WRITE (3,130) '---------------', '-----', '-----', '-----',
     +              '-----', '-----', '-----'

C Loop for temperature data output. Check every day of every year to
C see if data has been recorded.

      DO 10 Y = 1, 36
         DO 20 M = 1, 12
            DO 30 D = 1, 31

C Calculate the Julian day, to be used in output

               CALL JULIAN (JDAY, M, D

C Write out temperatures if data exists for the day

               IF (FLAG(M,D,Y) .EQ. 1) THEN
                  WRITE (3,140) M, D, Y+55, JDAY,
     +            AMAX(M,D,Y), AMIN(M,D,Y), AMEAN(M,D,Y),
     +            SMAX(M,D,Y), SMIN(M,D,Y), SMEAN(M,D,Y)
               ENDIF

C Clear arrays for analysis of next stream

               FLAG(M,D,Y)  = 0
               AMAX(M,D,Y)  = 0.0
               AMIN(M,D,Y)  = 0.0
               AMEAN(M,D,Y) = 0.0
               SMAX(M,D,Y)  = '      '
               SMIN(M,D,Y)  = '      '
               SMEAN(M,D,Y) = '      '

   30       CONTINUE
   20    CONTINUE
   10 CONTINUE

  100 FORMAT (A14, 1X, A)
  110 FORMAT (A)
  120 FORMAT (19X, A16, 11X, A18)
  130 FORMAT (A15, 5X, A5, 3X, A5, 3X, A5, 5X,
     +        A5, 3X, A5, 3X, A5)
  140 FORMAT (I2.2, '/', I2.2, '/', I2, 4X, I3, 4X,
     +        F6.2, 2X, F6.2, 2X, F6.2, 4X, A6, 2X, A6, 2X, A6)
      RETURN
      END


C ***********************************************************************
      SUBROUTINE JULIAN (JDAY, M, D)
C ***********************************************************************

C Purpose
C Determines the Julian day (1-365) for a given day out of the year.

C Description of variables
C Input
C M, D: Month and day of desired date
C
C Ouput
C JDAY: Julian day; ranges from 1 (January 1) to 365 (December 31)

C Declarations for calling arguments
      INTEGER *2 JDAY
      INTEGER M, D

C Determine the Julian day. This is done by finding the number of days
C elapsed in the preceding months, then adding this to the day number in
C the current month.

      IF (M .EQ. 1) THEN
         JDAY = D
      ELSEIF (M .EQ. 2) THEN
         JDAY = 31 + D
      ELSEIF (M .EQ. 3) THEN
         JDAY = 59 + D
      ELSEIF (M .EQ. 4) THEN
         JDAY = 90 + D
      ELSEIF (M .EQ. 5) THEN
         JDAY = 120 + D
      ELSEIF (M .EQ. 6) THEN
         JDAY = 151 + D
      ELSEIF (M .EQ. 7) THEN
         JDAY = 181 + D
      ELSEIF (M .EQ. 8) THEN
         JDAY = 212 + D
      ELSEIF (M .EQ. 9) THEN
         JDAY = 243 + D
      ELSEIF (M .EQ. 10) THEN
         JDAY = 273 + D
```

```
      ELSEIF (M .EQ. 11) THEN
         JDAY = 304 + D
      ELSEIF (M .EQ. 12) THEN
         JDAY = 334 + D
      ENDIF

      RETURN
      END


C ***********************************************************************
      SUBROUTINE RELATE (MIDFN, OUTFN, TYPE)
C ***********************************************************************

C Purpose
C Determines mean temperatures on the given time scale to be used for
C correlation by the next program in the sequence. This temperature data
C is written out to the output file, showing air and stream temperatures
C side by side for each day.

C Description of variables
C Input
C MIDFN: Intermediate filename, used for input
C OUTFN: Name of output file with mean temperatures
C TYPE:  Time scale for correlation
C        (1: daily; 2: weekly; 3: monthly; 4: yearly)
C
C Local
C SSTN:          Stream temperature station name
C ASTN:          True air temperature station name
C RECORD:        Dummy variable for reading from files
C FDATE, LDATE: First and last date in a time period
C MO, DAY, YR:  Date read from intermediate file
C YRPRV:         Previous year read in
C X:             Number of days in entire time period
C N:             Day number within time period
C MAX:           Maximum number of days to be considered in time period
C JDAY:          Julian day
C FJDAY          First Julian day in time period
C A, S:          Number of air and stream data points
C AMAX, AMIN, AMEAN: Air temperature data for time period
C SMAX, SMIN, SMEAN: Stream temperature data for time period
C AMTOT, SMTOT: Total of air, stream mean temperatures summed together
C I:             Counter variable
C WRITN:         Flag indicating whether data has been written out

C Subroutines called
C RLHEAD: Creates header for output file
C DYTIME: Determines essential information for daily time scale
C WKTIME: Determines essential information for weekly time scale
C MOTIME: Determines essential information for monthly time scale
C YRTIME: Determines essential information for yearly time scale
C RLOUT:  Writes temperature data to output file

C Declarations for calling arguments
      CHARACTER *12 MIDFN, OUTFN
      INTEGER TYPE

C Local declarations
      CHARACTER SSTN *48, ASTN *48, RECORD *1
      CHARACTER *8 FDATE, LDATE
      INTEGER MO, DAY, YR, YRPRV, X, N, MAX
      INTEGER *2 JDAY, FJDAY
      REAL A, AMAX, AMIN, AMEAN, AMTOT
      REAL S, SMAX, SMIN, SMEAN, SMTOT
      INTEGER I, WRITN

C Open appropriate files based on the desired time scale

      OPEN (UNIT=1, FILE=('STREAMS\'//MIDFN))
      IF (TYPE .EQ. 1) THEN
         OPEN (UNIT=2, FILE=('DAILY\'//OUTFN))
      ELSEIF (TYPE .EQ. 2) THEN
         OPEN (UNIT=2, FILE=('WEEKLY\'//OUTFN))
      ELSEIF (TYPE .EQ. 3) THEN
         OPEN (UNIT=2, FILE=('MONTHLY\'//OUTFN))
      ELSEIF (TYPE .EQ. 4) THEN
         OPEN (UNIT=2, FILE=('YEARLY\'//OUTFN))
      ENDIF
```

```
C Read in station names; create output file header

      READ (1,100) SSTN
      READ (1,100) ASTN
      DO 10 I = 1, 4
         READ (1,110) RECORD
   10 CONTINUE

      CALL RLHEAD (SSTN, ASTN, TYPE, X

C Read in temperature data from first day, to begin read loop

      READ (1,120, END=99) MO, DAY, YR, JDAY,
     +                     AMAX, AMIN, AMEAN, SMAX, SMIN, SMEAN

C Loop to evaluate temperature data; only consider data from April to
C October inclusive

      DO WHILE (.TRUE.)
         DO WHILE ((JDAY .GE. 91) .AND. (JDAY .LE. 304))
            N = 1
            A = 0.0
            S = 0.0
            WRITN = 0
            AMTOT = 0.0
            SMTOT = 0.0

C Determine first and last date, first Julian day, and maximum days to
C consider for the time period, based on the given time scale

            IF (TYPE .EQ. 1) THEN
               CALL DYTIME (MO, DAY, YR, JDAY, FDATE, FJDAY, MAX)
            ELSEIF (TYPE .EQ. 2) THEN
               CALL WKTIME (YR, JDAY, FDATE, LDATE, FJDAY, MAX)
            ELSEIF (TYPE .EQ. 3) THEN
               CALL MOTIME (MO, DAY, YR, FDATE, LDATE, FJDAY, MAX)
            ELSEIF (TYPE .EQ. 4) THEN
               CALL YRTIME (YR, JDAY, FDATE, LDATE, MAX)
            ENDIF

C Continue reading temperatures until end of time period

            DO WHILE ((N .LE. MAX) .AND. (JDAY .LE. 304))

C Calculate mean stream temperature if it is not explicitly given;
C increment number of stream data points

               IF (SMEAN .EQ. 0.0) THEN
                  IF ((SMAX .NE. 0.0) .AND. (SMIN .NE. 0.0)) THEN
                     SMEAN = 0.5 * (SMAX + SMIN)
                     S = S + 1.0
                  ENDIF
               ELSE
                  S = S + 1.0
               ENDIF

C Increment values, including number of air data points, and update mean
C temperature totals. This can only be done if the mean air temperature
C is above freezing, since ice cover may become a factor.

               N = N + 1
               YRPRV = YR
               IF (AMEAN .GT. 0) THEN
                  A = A + 1.0
                  AMTOT = AMTOT + AMEAN
                  SMTOT = SMTOT + SMEAN
               ELSE
                  S = S - 1.0
               ENDIF

C Read next temperatures available, continuing loop

               READ (1,120, END=99) MO, DAY, YR, JDAY
     +                              AMAX, AMIN, AMEAN, SMAX, SMIN, SMEAN

C If the date of this new data is not within the same time period,
C force exiting of time period loop

               IF ((JDAY - FJDAY .GE. MAX) .OR. (YR .NE. YRPRV)) THEN
```

A-8

```
            N = MAX + 1
         ENDIF
      ENDDO

C Write out mean temperatures if data is present for the majority of the
C days in the time period.

      IF (S .GT. X/2) THEN
         CALL RLOUT (FDATE, LDATE, FJDAY, YR, A, AMTOT, S, SMTOT,
     +               TYPE)
         WRITN = 1
      ENDIF

   ENDDO

C Read first temperature data for next time period

   READ (1,120, END=99) MO, DAY, YR, JDAY,
     +                  AMAX, AMIN, AMEAN, SMAX, SMIN, SMEAN
 ENDDO

 99 CONTINUE

C Arriving at the end of this loop may have interrupted a time period.
C This data is written out, if necessary. Close all open files.

      IF ((WRITN .NE. 1) .AND. (S .GT. X/2)) THEN
         CALL RLOUT (FDATE, LDATE, FJDAY, YR, A, AMTOT, S, SMTOT, TYPE)
      ENDIF

      CLOSE (UNIT=1)
      CLOSE (UNIT=2)

  100 FORMAT (15X, A48)
  110 FORMAT (A)
  120 FORMAT (I2, 1X, I2, 1X, I2, 4X, I3, 4X,
     +        F6.2, 2X, F6.2, 2X, F6.2, 4X, F6.2, 2X, F6.2, 2X, F6.2)

      RETURN
      END


C *********************************************************************
      SUBROUTINE RLHEAD (SSTN, ASTN, TYPE, X)
C *********************************************************************

C Purpose
C Creates header for output file. Also returns the number of days in the
C entire time period being considered.

C Description of variables
C Input
C SSTN: Stream temperature station name
C ASTN: Air temperature station name
C TYPE: Time scale for correlation
C       (1: daily; 2: weekly; 3: monthly; 4: yearly)
C
C Output
C X:    Number of days in entire time period

C Declarations for calling arguments
      CHARACTER SSTN *48, ASTN *48
      INTEGER TYPE, X

C Create file header

      WRITE (2,100) 'Stream data:  ', SSTN
      WRITE (2,100) 'Air temp data:', ASTN

C Determine the number of days in the entire time period

      IF (TYPE .EQ. 1) THEN
         WRITE (2,110) 'Type of correlation: DAILY'
         X = 1
      ELSEIF (TYPE .EQ. 2) THEN
         WRITE (2,110) 'Type of correlation: WEEKLY'
         X = 7
      ELSEIF (TYPE .EQ. 3) THEN
         WRITE (2,110) 'Type of correlation: MONTHLY'
         X = 31
      ELSEIF (TYPE .EQ. 4) THEN
         WRITE (2,110) 'Type of correlation: YEARLY'
         X = 214
      ENDIF

      WRITE (2,110) ' '

C Create table header in file

      IF (TYPE .EQ. 1) THEN
         WRITE (2,120) 'Date       Julian', 'Air   ', 'Stream'
         WRITE (2,120) '-----------------', '------', '------'
      ELSEIF (TYPE .EQ. 4) THEN
         WRITE (2,120) 'Date Range               Year', 'Air   ','Stream'
         WRITE (2,120) '-----------------------------', '------','------'
      ELSE
         WRITE (2,120) 'Date Range             Julian', 'Air   ','Stream'
         WRITE (2,120) '-----------------------------', '------','------'
      ENDIF

  100 FORMAT (A14, 1X, A48)
  110 FORMAT (A)
  120 FORMAT (A, 6X, A6, 6X, A6)

      RETURN
      END


C *********************************************************************
      SUBROUTINE RLOUT (FDATE, LDATE, FJDAY, YR, A,AMTOT, S,SMTOT, TYPE)
C *********************************************************************

C Purpose
C Writes out mean air and stream temperatures for the given time period
C in the appropriate output file for the time scale. These files are
C used by the next program in the sequence to determine the correlation
C coefficients.

C Description of variables
C Input
C FDATE, LDATE: First and last date in the time period
C FJDAY         First Julian day in time period
C A, S:         Number of air and stream data points
C AMTOT, SMTOT: Total of air, stream mean temperatures summed together
C YR:           Year of the time period
C TYPE:         Time scale for correlation
C               (1: daily; 2: weekly; 3: monthly; 4: yearly)

C Declarations for caling arguments
      CHARACTER *8 FDATE, LDATE
      INTEGER *2 FJDAY
      REAL A, AMTOT, S, SMTOT
      INTEGER YR, TYPE

C Write out data in a format specific to the type of time scale. Daily
C and yearly time scales, for instance, are less interested in some
C dates than others. Note that the mean temperatures for the period
C are calculated "on the fly."

      IF (TYPE .EQ. 1) THEN
         WRITE (2,100) FDATE, FJDAY, AMTOT/S, SMTOT/S
      ELSEIF (TYPE .EQ. 4) THEN
         WRITE (2,110) FDATE, LDATE, 1900+YR, AMTOT/A, SMTOT/S
      ELSE
         WRITE (2,110) FDATE, LDATE, FJDAY, AMTOT/A, SMTOT/S
      ENDIF

  100 FORMAT (A8, 5X, I4, 6X, F6.2, 6X, F6.2)
  110 FORMAT (A8, ' - ', A8, 5X, I4, 6X, F6.2, 6X, F6.2)

      RETURN
      END


C *********************************************************************
      SUBROUTINE DYTIME (MO, DAY, YR, JDAY, FDATE, FJDAY, MAX)
C *********************************************************************
```

```
C Purpose
C Converts the given date to a character format to be used in output.
C Also returns the Julian day and the total number of days in the time
C period (1).

C Description of variables
C Input
C MO, DAY, YR: Date given, read from file
C JDAY:        Julian day, read from file
C
C Output
C FDATE:       First (and only) date in time period
C FJDAY        First (and only) Julian day in time perio
C MAX:         Maximum number of days to be considered (1)

C Declarations of calling arguments
      CHARACTER *8 FDATE
      INTEGER MO, DAY, YR, MAX
      INTEGER *2 JDAY, FJDAY

C Assign Julian day, maximum number of days to consider

      FJDAY = JDAY
      MAX = 1

C Convert date to a character format

      FDATE(1:1) = CHAR(MOD(MO/10, 10) + 48)
      FDATE(2:2) = CHAR(MOD(MO, 10) + 48)
      FDATE(3:3) = '/'
      FDATE(4:4) = CHAR(MOD(DAY/10, 10) + 48)
      FDATE(5:5) = CHAR(MOD(DAY, 10) + 48
      FDATE(6:6) = '/'
      FDATE(7:7) = CHAR(MOD(YR/10, 10) + 48)
      FDATE(8:8) = CHAR(MOD(YR, 10) + 48)

      RETURN
      END


C ***********************************************************************
      SUBROUTINE WKTIME (YR, JDAY, FDATE, LDATE, FJDAY, MAX)
C ***********************************************************************

C Purpose
C Converts the given date to a character format to be used in output.
C Also returns the first and last dates of the time period, the first
C Julian day, and the total number of days.

C Description of variables
C Input
C YR:           Year given, read from fil
C JDAY:         Julian day, read from fil
C
C Output
C FDATE, LDATE: First and last date in time period
C FJDAY         First Julian day in time period
C MAX:          Maximum number of days to be considered
C
C Local
C CFMO, CLMO:   First and last month in time period, as characters
C CFDY, CLDY:   First and last day in time period, as characters
C CYR:          Year of the time period, as character
C M(i), D(i):   First and last month and day, as integer

C Declarations of calling arguments
      CHARACTER *8 FDATE, LDATE
      INTEGER YR, MAX
      INTEGER *2 JDAY, FJDAY

C Local declarations
      CHARACTER *2 CFMO, CLMO, CFDY, CLDY, CYR
      INTEGER M(2), D(2)

C Set initial values; a repeated process is used to determine the true
C first and last days of the time period.

      M(1) = 4
      D(1) = 1
      M(2) = 4
      D(2) = 7
      FJDAY = 91

C Increment the variable D by one week until correct value is found

      DO WHILE (FJDAY .LE. JDAY-7)
         D(1) = D(1) + 7
         D(2) = D(2) + 7

C After coming to end of a month in the check, start the next month:
C reset D and increment M.

         DO 10 I = 1, 2
            IF ( ((M(I) .EQ. 4) .OR. (M(I) .EQ. 6) .OR. (M(I) .EQ. 9))
     +                   .AND. (D(I) .GT. 30)) THEN
               M(I) = M(I) + 1
               D(I) = D(I) - 30
            ELSEIF ( ((M(I) .EQ. 5) .OR. (M(I) .EQ. 7) .OR. (M(I).EQ.8))
     +                       .AND. (D(I) .GT. 31)) THEN
               M(I) = M(I) + 1
               D(I) = D(I) - 31
            ELSEIF ((M(I) .EQ. 10) .AND. (D(I) .GT. 31)) THEN
               D(I) = 31
            ENDIF
   10    CONTINUE

C Increment the Julian day by one week and continue the loop

         FJDAY = FJDAY + 7
      ENDDO

C Since day #91 (April 1) is divisible by 7, we can use the followin
C expression to determine the number of days remaining in the given week

      MAX = 7 - MOD(JDAY, 7)

C Convert month, day, and year to characters; assign the first and last
C date of the time period

      CFMO(1:1) = CHAR(MOD(M(1)/10, 10) + 48)
      CFMO(2:2) = CHAR(MOD(M(1), 10) + 48)
      CLMO(1:1) = CHAR(MOD(M(2)/10, 10) + 48)
      CLMO(2:2) = CHAR(MOD(M(2), 10) + 48)

      CFDY(1:1) = CHAR(MOD(D(1)/10, 10) + 48)
      CFDY(2:2) = CHAR(MOD(D(1), 10) + 48)
      CLDY(1:1) = CHAR(MOD(D(2)/10, 10) + 48)
      CLDY(2:2) = CHAR(MOD(D(2), 10) + 48)

      CYR(1:1) = CHAR(MOD(YR/10, 10) + 48)
      CYR(2:2) = CHAR(MOD(YR, 10) + 48

      FDATE = CFMO // '/' // CFDY // '/' // CYR
      LDATE = CLMO // '/' // CLDY // '/' // CYR

      RETURN
      END


C ***********************************************************************
      SUBROUTINE MOTIME (MO, DAY, YR, FDATE, LDATE, FJDAY, MAX)
C ***********************************************************************

C Purpose
C Converts the given to date to a character format to be used in output.
C Also returns the first and last dates of the time period, the firs
C Julian day, and the total number of days.

C Description of variables
C MO, DAY, YR:  Date given, read from file
C JDAY:         Julian day, read from file
C
C Output
C FDATE, LDATE:  First and last date in time period
C FJDAY          First Julian day in time period
C MAX:           Maximum number of days to be considered
C
```

```
C Local
C CDY, CMO, CYR: Month, day, and year of time period, as characters
C D:            First day of month

C Declarations of calling arguments
      CHARACTER *8 FDATE, LDATE
      INTEGER MO, DAY, YR, MAX
      INTEGER *2 FJDAY

C Local declarations
      CHARACTER CDY *4, CMO *2, CYR *2
      INTEGER D

C Find Julian day for the first day of the month

      D = 1
      CALL JULIAN (FJDAY, MO, D)

C Determine last day of month, along with the number of days remaining

      IF ((MO .EQ. 4) .OR. (MO .EQ. 6) .OR. (MO .EQ. 9)) THEN
         CDY = '/30/'
         MAX = 30 - DAY + 1
      ELSEIF ((MO .EQ. 5) .OR. (MO .EQ. 7) .OR.
     +        (MO .EQ. 8) .OR. (MO .EQ. 10)) THEN
         CDY = '/31/'
         MAX = 31 - DAY + 1
      ENDIF

C Convert month and year to characters; assign the first and last date
C of the time period

      CMO(1:1) = CHAR(MOD(MO/10, 10) + 48
      CMO(2:2) = CHAR(MOD(MO, 10) + 48)
      CYR(1:1) = CHAR(MOD(YR/10, 10) + 48
      CYR(2:2) = CHAR(MOD(YR, 10) + 48)

      FDATE = CMO // '/01/' // CYR
      LDATE = CMO //  CDY   // CYR

      RETURN
      END


C ***********************************************************************
      SUBROUTINE YRTIME (YR, JDAY, FDATE, LDATE, MAX)
C ***********************************************************************

C Purpose
C Converts the given to date to a character format to be used in output
C Also returns the first and last dates of the time period, the first
C Julian day, and the total number of days.

C Description of variables
C Input
C YR:           Year given, read from fil
C JDAY:         Julian day, read from fil
C
C Output
C FDATE, LDATE: First and last date in time period
C MAX:          Maximum number of days to be considered
C
C Local
C CYR:          Year of the time period, as character

C Declarations of calling arguments
      CHARACTER *8 FDATE, LDATE
      INTEGER YR, MAX
      INTEGER *2 JDAY

C Local declarations
      CHARACTER *2 CYR

C Determine number of days remaining in the year

      MAX = 304 - JDAY + 1

C Convert year to a character; assign first and last date of the time
C period

      CYR(1:1) = CHAR(MOD(YR/10, 10) + 48)
      CYR(2:2) = CHAR(MOD(YR, 10) + 48

      FDATE = '04/01/' // CYR
      LDATE = '10/31/' // CYR

      RETURN
      END
```

```
C ***********************************************************************
      PROGRAM CORREL
C ***********************************************************************
C Author: John Pilgrim
C Date:   6/5/95

C Purpose
C The program reads in files containing both mean air and stream
C temperatures for a number of stations throughout Minnesota. These
C files were generated by the previous program in the sequence, SETUP,
C for four different time scales: daily, weekly, monthly, and yearly.
C Correlation coefficients are then determined for each stream using a
C linear least-squares regression model (Tw = A + B*Ta). Additional
C statistical parameters (R^2, Sp, and Se) are calculated as well, and
C all this information is written out to separate files for each stream
C Correlations are also done for all the data from all streams thrown
C together, and the results are written to a different output file. At
C this point, the program sequence and the correlation process is
C complete.

C Description of variables
C INFN(i,j):  Array of temperature input filenames
C OUTFN(i):   Array of correlation output filenames
C SUMMFN:     Name of output file summarizing all correlation data
C COORFN:     Name of file containing station coordinate:
C LATS(i), LONGS(i): Coordinates of stream temperature station
C LATA(i), LONGA(i): Coordinates of air temperature station
C SA(i,j), SB(i,j):  Correlation coefficients for each stream
C SRSQ(i,j):  Statistical parameter R^2 for each stream
C SNUM(i,j):  Number of data points used for each stream
C SNTOT(i,j): Number of possible data points within time range
C TNUM(i):    Total number data points for all MN stream:
C TSUMA(i):   Sum of all air temperature values; for all MN streams
C TSQA(i):    Sum of squares of air temperature values; for all streams
C TSUMW(i):   Sum of all water temperature values; for all MN streams
C TPROD(i):   Sum of products of air and water temperature values
C TNTOT(i):   Total number of possible data points for all MN streams
C N:          Counter variable; for stream identification (1-43)

C Subroutines called
C ASSIGN: Assigns filenames to the arrays
C COORDS: Reads in station coordinates from a separate, custom file
C RELATE: Performs general correlation analysis for each stream on all
C         time scales
C OVRALL: Performs overall Minnesota correlation analysi:
C SUMMRY: Write out correlation data for each stream

C Variable declarations
      CHARACTER INFN(43,4) *20, OUTFN(43) *12, SUMMFN *12, COORFN *18
      REAL LATS(43), LONGS(43), LATA(43), LONGA(43)
      REAL SA(43,4), SB(43,4), SRSQ(43,4)
      INTEGER *4 SNUM(43,4), SNTOT(43,4)
      REAL TNUM(4), TSUMA(4), TSQA(4), TSUMW(4), TPROD(4
      INTEGER *4 TNTOT(4)
      INTEGER N

C Assign filenames

      DATA SUMMFN / 'SUMMARY.DAT' /
      DATA COORFN / 'COORDS.DAT' /

      CALL ASSIGN (INFN, OUTFN)

C Read in air and stream station coordinates from separate file

      CALL COORDS (COORFN, LATS, LONGS, LATA, LONGA)

      WRITE (*,*) ' '

C Perform correlation analysis. Determine correlation coefficients and
C statistical parameters for each stream; find quantities to be used fo:
C overall Minnesota correlation.

      DO 10 N = 1, 43
         WRITE (*,*) OUTFN(N)
         CALL RELATE (INFN, OUTFN, LATS, LONGS, LATA, LONGA, N,
     +                SA, SB, SNUM, SNTOT, SRSQ,
     +                TNUM, TNTOT, TSUMA, TSQA, TSUMW, TPROD)
   10 CONTINUE

C Perform analysis for overall Minnesota correlation; write out to file

      WRITE (*,*) SUMMFN
      CALL OVRALL (INFN, SUMMFN, TNUM, TNTOT, TSUMA, TSQA, TSUMW, TPROD)

C Write out correlation data for each stream on all time scales

      CALL SUMMRY (LATS, LONGS, SA, SB, SNUM, SNTOT, SRSQ)

      END


C ***********************************************************************
      SUBROUTINE ASSIGN (INFN, OUTFN)
C ***********************************************************************

C Purpose
C Assigns names of data files into arrays. This is done for both the
C input files, on different time scales, and the output files.

C Description of variables
C Output
C INFN(i,j): Array of temperature input filenames
C OUTFN(i):  Array of correlation output filenames
C
C Local
C NUM: Stream index represented as character
C I:   Counter variable

C Declarations for calling arguments
      CHARACTER INFN(43,4) *20, OUTFN(43) *12

C Local declarations
      CHARACTER NUM *2
      INTEGER I

C Assign filenames to arrays. Convert the 2-digit number I to character

      DO 10 I = 1, 43
         NUM(1:1) = CHAR(MOD(I/10, 10) + 48)
         NUM(2:2) = CHAR(MOD(I, 10) + 48)

C Concatenate strings to create filenames

         INFN(I,1) = 'DAILY\COR'   // NUM // '_DY.DAT
         INFN(I,2) = 'WEEKLY\COR'  // NUM // '_WK.DAT
         INFN(I,3) = 'MONTHLY\COR' // NUM // '_MO.DAT
         INFN(I,4) = 'YEARLY\COR'  // NUM // '_YR.DAT
         OUTFN(I)  = 'RESULT' // NUM // '.DAT'
   10 CONTINUE

      RETURN
      END


C ***********************************************************************
      SUBROUTINE COORDS (COORFN, LATS, LONGS, LATA, LONGA)
C ***********************************************************************

C Purpose
C Reads in the latitudinal and longitudinal coordinates of both the
C air and stream stations used in measuring the temperature of a
C particular stream. These coordinates are later written to the fina:
C output file for a clear comparison. The input file must be custom-
C written to work with this program.

C Description of variables
C Input
C COORFN:            Name of file containing station coordinates
C LATS(i), LONGS(i): Coordinates of stream temperature station
C LATA(i), LONGA(i): Coordinates of air temperature station
C
C Local
C I:         Counter variable
C WHOLE(i): Whole number part of air station coordinates
C MIN(i):    Fractional part of air station coordinates, in minutes

C Declarations for calling arguments
```

```
      CHARACTER COORFN *20
      REAL LATS(43), LONGS(43), LATA(43), LONGA(43)

C Local declarations
      INTEGER I, WHOLE(2), MIN(2)

C Open custom coordinate file; read value:

      OPEN (UNIT=1, FILE=COORFN)

      DO 10 I = 1, 43
         READ (1,100) LATS(I),LONGS(I), WHOLE(1),MIN(1),WHOLE(2),MIN(2)

C Since the air temperature files provided give the coordinates in
C minutes, they should be converted to decimal format for consistency.

         LATA(I)  = WHOLE(1) + MIN(1)/60.
         LONGA(I) = WHOLE(2) + MIN(2)/60.
   10 CONTINUE

      CLOSE (UNIT=1)

  100 FORMAT (F8.5, 2X, F8.5, 2X, I2, 1X, I2, 2X, I2, 1X, I2)

      RETURN
      END


C ********************************************************************
      SUBROUTINE RELATE (INFN, OUTFN, LATS, LONGS, LATA, LONGA, N,
     +                   SA, SB, SNUM, SNTOT, SRSQ,
     +                   TNUM, TNTOT, TSUMA, TSQA, TSUMW, TPROD)
C ********************************************************************

C Purpose
C Performs correlation analysis on a set of data. A linear least-square:
C regression model (Tw = A + B*Ta) is used to determine a pair of
C correlation coefficients. Additional statistical parameters are
C calculated to determine the consistency and precision of the
C correlation. All these values are returned to the main program for
C later use along with running sums of necessary parameters.

C Description of variables
C Input
C INFN(i,j):  Array of temperature input filenames
C OUTFN(i):   Array of correlation output filenames
C LATS(i), LONGS(i): Coordinates of stream temperature station
C LATA(i), LONGA(i): Coordinates of air temperature station
C N:          Stream identification number (1-43)
C
C Output
C SA(i,j), SB(i,j):  Correlation coefficients for each stream
C SRSQ(i,j):  Statistical parameter R^2 for each stream
C SNUM(i,j):  Number of data points used for each stream
C SNTOT(i,j): Number of possible data points within time range
C TNUM(i):    Total number data points for all MN stream:
C TSUMA(i):   Sum of all air temperature values; for all MN streams
C TSQA(i):    Sum of squares of air temperature values; for all streams
C TSUMW(i):   Sum of all water temperature values; for all MN streams
C TPROD(i):   Sum of products of air and water temperature values
C TNTOT(i):   Total number of possible data points for all MN streams
C
C Local
C SSTN:          Stream temperature station name
C ASTN:          True air temperature station name
C RECORD:        Dummy variable for reading from files
C STRT:          First date read in from file
C FDATE, LDATE:  First and last date in file range
C FYEAR, LYEAR:  First and last year in file range
C PRE, POST:     Used to calculate number of days in short year
C JDAY:          Julian day
C FJDAY, LJDAY   First and last Julian day in file range
C TW, TA:        Mean stream and air temperatures read in from file
C A(i), B(i):    Linear correlation coefficients
C RSQ(i):        Statistical parameter R^2
C SP(i), SE(i):  Statistical parameters Sp, Se
C NUM(i):        Number of data points used in correlatio
C SUMA(i):       Sum of all air temperature values
C SQA(i):        Sum of squares of air temperature values
C SUMW(i):       Sum of all water temperature values
C PROD(i):       Sum of products of air and water temperature values
C NTOT(i):       Number of possible data points within time range
C I:             Counter variable
C FLAG:          Indicates whether start date has been read in
C NF, NL:        Numbers of first and last files used in correlation

C Subroutines called
C GTSTAT: Determines correlation coefficients and statistical parameters
C RLOUT:  Writes correlation data to output file

C Declarations for calling arguments
      CHARACTER INFN(43,4) *20, OUTFN(43) *12
      REAL LATS(43), LONGS(43), LATA(43), LONGA(43)
      INTEGER N
      REAL SA(43,4), SB(43,4), SRSQ(43,4)
      INTEGER *4 SNUM(43,4), SNTOT(43,4)
      REAL TNUM(4), TSUMA(4), TSQA(4), TSUMW(4), TPROD(4)
      INTEGER *4 TNTOT(4)

C Local declarations
      CHARACTER SSTN *48, ASTN *48, RECORD *1
      CHARACTER *6 STRT, FDATE, LDATE
      INTEGER YR, FYEAR, LYEAR, PRE, POST
      INTEGER *2 JDAY, FJDAY, LJDAY
      REAL TW, TA, A(4), B(4), RSQ(4), SP(4), SE(4)
      REAL NUM(4), SUMA(4), SQA(4), SUMW(4), PROD(4)
      INTEGER *4 NTOT(4)
      INTEGER I, FLAG, NF, NL

C Clear arrays; set initial values

      DO 10 I = 1, 4
         NUM(I)  = 0.0
         SUMA(I) = 0.0
         SQA(I)  = 0.0
         SUMW(I) = 0.0
         PROD(I) = 0.0
         NTOT(I) = 0.0
   10 CONTINUE

      FJDAY = 0
      LJDAY = 0
      FYEAR = 0
      LYEAR = 0

      FLAG = 0

C Open files and read in station names

      DO 20 I = 1, 4
         OPEN (UNIT=I, FILE=INFN(N,I))
   20 CONTINUE
      OPEN (UNIT=5, FILE=('CORRELS\'//OUTFN(N)) )

      DO 30 I = 1, 4
         READ (I,100) SSTN
         READ (I,100) ASTN
         DO 40 J = 1, 4
            READ (I,110) RECORD
   40    CONTINUE
   30 CONTINUE

C Loop for reading from input files, for each time scale

      DO 50 I = 1, 4
         DO WHILE (.TRUE.)

C Read in mean air and stream temperatures. The dates from the daily
C time scale are used to find a broad date range of the correlation.

            IF (I .EQ. 1) THEN
               READ (I,120, END=50) STRT, YR, JDAY, TA, TW

C If date read is the first one, assign accordingly; otherwise, trea
C as the last date seen thus far

               IF (FLAG .EQ. 0) THEN
                  FDATE = STRT
                  FJDAY = JDAY
```

```
                FYEAR = YR
                FLAG  = 1
             ENDIF
             LDATE = STRT
             LJDAY = JDAY
             LYEAR = YR
          ELSE
             READ (I,130, END=50) TA, TI
          ENDIF

C Increment statistical parameters

          NUM(I)  = NUM(I)  + 1.0
          SUMA(I) = SUMA(I) + TA
          SQA(I)  = SQA(I)  + TA*TA
          SUMW(I) = SUMW(I) + TW
          PROD(I) = PROD(I) + TA*TW

       ENDDO
   50 CONTINUE

C Close input files before calling subroutine, as the subroutine opens
C its own files based on NF and NL. Since this part of the program deal:
C with one stream, make sure correlation is done for stream no. N only
C (from N to N). Get correlation coefficients and statistical parameter:

      DO 60 I = 1, 4
         CLOSE (UNIT=I)
   60 CONTINUE

      NF = N
      NL = N
      CALL GTSTAT (A, B, RSQ, SP, SE, NUM, SUMA, SQA, SUMW, PROD,
     +             INFN, NF, NL)

C Calculate the total number of data points in the time range examined.
C Assume 31 weeks between April and October inclusive and an average 30
C days per month.
C In the first case, the time range is less than one year. To ensure
C validity, LJDAY must have some value other than its initial value (0)

      IF (LYEAR .EQ. FYEAR) THEN
         IF (LJDAY .NE. 0) THEN
            NTOT(1) = LJDAY - FJDAY + 1
            NTOT(2) = NTOT(1) / 7.0
            NTOT(3) = NTOT(1) / 30.0
            NTOT(4) = LYEAR - FYEAR + 1
         ENDIF

C If the time range is longer than one year, calculate the number of
C time periods contained within the full "middle" years and add to the
C number in the short "outside" years.

      ELSE
         PRE  = 304 - FJDAY + 1
         POST = LJDAY - 91 + 1
         NTOT(1) = PRE + 214*(LYEAR-FYEAR-1) + POST
         NTOT(2) = (PRE+3.0)/7.0 + 31.0*(LYEAR-FYEAR-1) + POST/7.0
         NTOT(3) = PRE/30.0 + 7.0*(LYEAR-FYEAR-1) + POST/30.0
         NTOT(4) = LYEAR - FYEAR + 1
      ENDIF

C Write header to output file. Include complete date range if any data
C was successfully correlated

      WRITE (5,140) 'Stream data:  ', SSTN, LATS(N), LONGS(N)
      WRITE (5,140) 'Air temp data:', ASTN, LATA(N), LONGA(N)

      IF (LYEAR .NE. 0) THEN
         WRITE (5,150) 'Date Range:   ', FDATE, FYEAR, LDATE, LYEAR
      ELSE
         WRITE (5,110) 'Date Range:   '
      ENDIF

C Write out correlation coefficients and statistical parameters to
C separate file for the stream

      CALL RLOUT (A, B, NUM, NTOT, RSQ, SP, SE)

C Update overall MN stream parameters; assign correlation coefficients
C to special array for the present stream; close output file

      DO 70 I = 1, 4
         TNUM(I)  = TNUM(I)  + NUM(I)
         TNTOT(I) = TNTOT(I) + NTOT(I)
         TSUMA(I) = TSUMA(I) + SUMA(I)
         TSQA(I)  = TSQA(I)  + SQA(I)
         TSUMW(I) = TSUMW(I) + SUMW(I)
         TPROD(I) = TPROD(I) + PROD(I)

         SA(N,I)    = A(I)
         SB(N,I)    = B(I)
         SNUM(N,I)  = INT4(NUM(I))
         SNTOT(N,I) = NTOT(I)
         SRSQ(N,I)  = RSQ(I)
   70 CONTINUE

      CLOSE (UNIT=5)

  100 FORMAT (15X, A48)
  110 FORMAT (A)
  120 FORMAT (A6, I2, 5X, I4, 6X, F6.2, 6X, F6.2)
  130 FORMAT (34X, F6.2, 6X, F6.2)
  140 FORMAT (A14, 1X, A48, 2X, F6.3, ', ', F6.3)
  150 FORMAT (A14, 1X, A6, I2, ' - ', A6, I2)

      RETURN
      END


C ***********************************************************************
      SUBROUTINE GTSTAT (A, B, RSQ, SP, SE, NUM, SUMA, SQA, SUMW, PROD,
     +                   INFN, NF, NL)
C ***********************************************************************

C Purpose
C Calculates the correlation coefficients and statistical parameters of
C interest, given a few necessary values. A general linear least-squares
C regression model (Tw = A + B*Ta) is used.

C Description of variables
C Input
C NUM(i):       Number of data points used in correlation
C SUMA(i):      Sum of all air temperature values
C SQA(i):       Sum of squares of air temperature values
C SUMW(i):      Sum of all water temperature values
C PROD(i):      Sum of products of air and water temperature values
C INFN(i,j):    Array of temperature input filenames
C NF, NL:       Numbers of first and last files used in correlation
C
C Output
C A(i), B(i):   Linear correlation coefficients
C RSQ(i):       Statistical parameter R^2
C SP(i), SE(i): Statistical parameters Sp, Se
C
C Local
C RECORD:       Dummy variable for reading from files
C STRT:         First date read in from file
C TW, TA:       Mean stream and air temperatures read in from file
C TPRED:        Stream temperature, predicted using coefficients A, I
C YBAR(i):      Mean stream temperature over entire file range
C SUMSP(i), SUMSE(i): Statistical parameters used to find Sp, Se
C N:            Counter variable; for stream identification (1-43)
C I, J:         Counter variables

C Declarations for calling arguments
      REAL A(4), B(4), RSQ(4), SP(4), SE(4)
      REAL NUM(4), SUMA(4), SQA(4), SUMW(4), PROD(4)
      CHARACTER INFN(43,4) *20
      INTEGER NF, NL

C Local declarations
      CHARACTER RECORD *1
      REAL TW, TA, TPRED
      REAL YBAR(4), SUMSP(4), SUMSE(4)
      INTEGER N, I, J

C Clear arrays; set initial values
```

```
      DO 10 I = 1, 4
         A(I)     = 0.0
         B(I)     = 0.0
         RSQ(I)   = 0.0
         SP(I)    = 0.0
         SE(I)    = 0.0
         SUMSP(I) = 0.0
         SUMSE(I) = 0.0
   10 CONTINUE

C Calculate correlation coefficients by least-squares regression.
C Expressions for the coefficients were determined previously with
C algebra. Also find the mean stream temperature over the time scale.

      DO 20 I = 1, 4
         IF (NUM(I) .GE. 4) THEN
            B(I) = ( PROD(I) - SUMA(I)*SUMW(I) / NUM(I) ) /
     +                ( SQA(I)  - SUMA(I)*SUMA(I) / NUM(I)
            A(I) = SUMW(I) / NUM(I) - SUMA(I)/NUM(I) * B(I)
            YBAR(I) = SUMW(I) / NUM(I)
         ENDIF
   20 CONTINUE

C Read from input files for each time scale to determine statistical
C parameters. If called from RELATE, this is done only for the present
C stream; if called from OVRALL, for all streams.

      DO 30 N = NF, NL
         DO 40 I = 1, 4
            OPEN (UNIT=I, FILE=INFN(N,I))
            DO 50 J = 1, 6
               READ (I,110) RECORD
   50       CONTINUE
   40    CONTINUE

C Read in temperature values; calculate a predicted stream temperature
C using the model, used in finding statistical parameter:

         DO 60 I = 1, 4
            DO WHILE (.TRUE.)
               IF (I .EQ. 1) THEN
                  READ (I,120, END=60) TA, TW
               ELSE
                  READ (I,130, END=60) TA, TW
               ENDIF

               IF (NUM(I) .GE. 4) THEN
                  TPRED = A(I) + B(I)*TA
                  SUMSP(I) = SUMSP(I) + (TW - TPRED)*(TW - TPRED)
                  SUMSE(I) = SUMSE(I) + (TW - YBAR(I))*(TW - YBAR(I))
               ENDIF
            ENDDO
   60    CONTINUE
   30 CONTINUE

C Calculate statistical parameters R^2, Sp, Se according to definitions

      DO 70 I = 1, 4
         IF (NUM(I) .GE. 4) THEN
            RSQ(I) = 1.0  - (SUMSP(I)/NUM(I)) / (SUMSE(I)/NUM(I))
            SP(I)  = SQRT(SUMSP(I) / NUM(I))
            SE(I)  = SQRT(SUMSE(I) / NUM(I))
         ENDIF
   70 CONTINUE

C Close input files opened by this subroutine

      DO 80 I = 1, 4
         CLOSE (UNIT=I)
   80 CONTINUE

  110 FORMAT (A)
  120 FORMAT (23X, F6.2, 6X, F6.2)
  130 FORMAT (34X, F6.2, 6X, F6.2)

      RETURN
      END
```

```
C ***********************************************************************
      SUBROUTINE RLOUT (A, B, NUM, NTOT, RSQ, SP, SE)
C ***********************************************************************

C Purpose
C Writes out the correlation coefficients and statistical parameters to
C a file. Data is presented in a concise tabular format, listing the
C time scales clearly.

C Description of variables
C Input
C A(i), B(i):   Linear correlation coefficients
C NUM(i):       Number of data points used in correlation
C RSQ(i):       Statistical parameter R^2
C SP(i), SE(i): Statistical parameters Sp, Se
C NTOT(i):      Number of possible data points within time range
C
C Local
C LABEL(i):     String for names of time scales
C I:            Counter variable

C Declarations for calling arguments
      REAL A(4), B(4), NUM(4), RSQ(4), SP(4), SE(4)
      INTEGER *4 NTOT(4)

C Local declarations
      CHARACTER LABEL(4) *7
      INTEGER I

C Create table header

      DATA LABEL / 'Daily', 'Weekly', 'Monthly', 'Yearly' /

      WRITE (5,110) ' '
      WRITE (5,110) 'Correlation model:   Tw = A + B*Ta'
      WRITE (5,110) ' '
      WRITE (5,120) 'Time scale', 'A    ', 'B    ', 'N      / Total',
     +                          'R^2  ', 'Sp   ', 'Se   '
      WRITE (5,120) '----------', '-----', '-----', '-------------',
     +                          '-----', '-----', '-----'

C Write out coefficients and parameters for each time scale

      DO 10 I = 1, 4
         WRITE (5,130) LABEL(I), A(I), B(I), INT4(NUM(I)), NTOT(I),
     +                 RSQ(I), SP(I), SE(I)
   10 CONTINUE

  110 FORMAT (A)
  120 FORMAT (A10, 5X, A5, 3X, A5, 5X, A13, 5X, A5, 3X, A5, 3X, A5)
  130 FORMAT (A7, 7X, F6.2, 2X, F6.3, 5X, I5, ' / ', I5, 4X,
     +        F6.3, 2X, F6.2, 2X, F6.2

      RETURN
      END


C ***********************************************************************
      SUBROUTINE OVRALL (INFN, SUMMFN, TNUM, TNTOT, TSUMA, TSQA, TSUMW,
     +                   TPROD)
C ***********************************************************************

C Purpose
C Performs a correlation analysis using the temperature data from al:
C the Minnesota streams thrown together. New, overall coefficients and
C parameters are determined.

C Description of variables
C Input
C INFN(i,j): Array of temperature input filenames
C SUMMFN:    Name of output file summarizing all correlation data
C TNUM(i):   Total number data points for all MN streams
C TSUMA(i):   Sum of all air temperature values; for all MN streams
C TSQA(i):    Sum of squares of air temperature values; for all streams
C TSUMW(i):   Sum of all water temperature values; for all MN stream:
C TPROD(i):   Sum of products of air and water temperature values
C TNTOT(i):  Total number of possible data points for all MN streams
C
C Local
```

```
C A(i), B(i):   Linear correlation coefficients
C NUM(i):       Number of data points used in correlatioı
C RSQ(i):       Statistical parameter R^2
C SP(i), SE(i): Statistical parameters Sp, Se
C NF, NL:       Numbers of first and last files to be used in correlation

C Subroutines called
C RLOUT: Writes correlation data to output file

C Declarations for calling arguments
      CHARACTER INFN(43,4) *20, SUMMFN *1:
      REAL TNUM(4), TSUMA(4), TSQA(4), TSUMW(4), TPROD(4
      INTEGER *4 TNTOT(4)

C Local declarations
      REAL A(4), B(4), RSQ(4), SP(4), SE(4)
      INTEGER NF, NL

C Open summary output file; perform correlation analysis on all streams
C (Nos. 1-43)

      OPEN (UNIT=5, FILE=SUMMFN)

      NF = 1
      NL = 43
      CALL GTSTAT (A, B, RSQ, SP, SE, TNUM, TSUMA, TSQA, TSUMW, TPROD,
     +             INFN, NF, NL)

C Write out results to top of file. The rest of the file will contain
C correlation results for the individual streams on different time
C scales.

      WRITE (5,110) 'Stream/Air Temperature Correlations:  Minnesota'

      CALL RLOUT (A, B, TNUM, TNTOT, RSQ, SP, SE)

  110 FORMAT (A)

      RETURN
      END


C ***********************************************************************
      SUBROUTINE SUMMRY (LATS, LONGS, SA, SB, SNUM, SNTOT, SRSQ)
C ***********************************************************************

C Purpose
C Writes out the final correlation results for individual streams. The
C results are sorted by time scale and presented in tabular form so tha
C visual comparison of the coefficients and parameters for each stream
C is possible. Determining these values fulfills the goal of the
C project.

C Description of variables
C Input
C LATS(i), LONGS(i): Coordinates of stream temperature station
C SA(i,j), SB(i,j):  Linear correlation coefficients
C SNUM(i,j):         Number of data points used in correlation
C SRSQ(i,j):         Statistical parameter R^2
C SNTOT(i):          Number of possible data points within time range
C
C Local
C LABEL(i): String for names of time scales
C I:        Counter variable

C Declarations for calling arguments
      REAL LATS(43), LONGS(43), SA(43,4), SB(43,4), SRSQ(43,4)
      INTEGER *4 SNUM(43,4), SNTOT(43,4)

C Local declarations
      CHARACTER LABEL(4) *7
      INTEGER N, I

C Create table header; repeat for each time scale studie

      DATA LABEL / 'DAILY', 'WEEKLY', 'MONTHLY', 'YEARLY' /

      DO 10 I = 1, 4
         WRITE (5,100) ' '
         WRITE (5,100) ' '
         WRITE (5,110) 'Time scale:', LABEL(I)
         WRITE (5,100) ' '
         WRITE (5,120) 'Stream No / Coords  ', 'A    ', 'B    ',
     +                 'N     / Total', 'R^2  '
         WRITE (5,120) '--------------------', '-----', '-----',
     +                 '-------------', '-----'

C Write out results to table if a correlation was successfully found.
C close summary output file

         DO 20 N = 1, 43
            IF ((SA(N,I) .NE. 0) .AND. (SB(N,I) .NE. 0)) THEN
               WRITE (5,130) N, LATS(N), LONGS(N), SA(N,I), SB(N,I),
     +                       SNUM(N,I), SNTOT(N,I), SRSQ(N,I)
            ENDIF
   20    CONTINUE
   10 CONTINUE

      CLOSE (UNIT=5)

  100 FORMAT (A)
  110 FORMAT (A11, 1X, A7)
  120 FORMAT (A20, 5X, A5, 3X, A5, 5X, A13, 5X, A5)
  130 FORMAT (1X, I2.2, 3X, F6.3, ', ', F6.3, 4X, F6.2, 2X, F6.3, 5X,
     +        I5, ' / ', I5, 4X, F6.3)

      RETURN
      END
```