Environment Canada Environnement Canada

Canada

NATIONAL WATER RESEARCH INSTITUTE

INSTITUT NATIONAL DE RECHERCHE SUR LES EAUX

CALCULATION OF BASE FLOW USING A RELATIONAL DATABASE MANAGEMENT SYSTEM APPROACH

Piggott, A.R., S. Moin and C. Southam

NWRI Contribution No. 02-026

CALCULATION OF BASE FLOW USING A RELATIONAL DATABASE MANAGEMENT SYSTEM APPROACH

Andrew R. Piggott [1], Syed Moin [2], and Chuck Southam [2]

[1] National Water Research Institute, 867 Lakeshore Road, Burlington, Ontario L7R 4A6
[2] Meteorological Service of Canada - Ontario Region, 867 Lakeshore Road, Burlington, Ontario L7R 4A6

National Water Research Institute Contribution Number 02-026

Abstract

An implementation of the United Kingdom Institute of Hydrology method for the calculation of base flow within a relational database management system context is reported. The approach is demonstrated using Structured Query Language and related code fragments applied to streamflow monitoring data that are cited in the documentation of the method. Arbitrary aspects of the implementation are assessed using streamflow data that are typical for southern Ontario. The implications of these aspects are estimated to be modest relative to yearly and monthly average base flows. The relational database compatibility and portability of this implementation may enable the application of the approach in a range of settings.

Résumé

On présente une méthode mise en oeuvre par l'Institute of Hydrology du Royaume-Uni pour le calcul du débit de base dans un environnement de système de gestion de bases de données relationnelles. On fait la démonstration de cette approche en utilisant un langage d'interrogation structuré et des extraits pertinents des programmes informatiques utilisés avec les données de surveillance du débit des cours d'eau mentionnées dans la documentation de la méthode. On évalue les aspects arbitraires de cette méthode en utilisant des données de débit de cours d'eau typiques du sud de l'Ontario. On estime que les conséquences de ces aspects sont modestes par rapport aux débits de base moyens annuels et mensuels. À cause de sa compatibilité et de sa portabilité pour les diverses bases de données relationnelles, cette approche convient peut-être à toute une gamme de conditions.

Introduction

Separation of streamflow monitoring data into base flow and direct surface runoff components is often useful in the characterization of the hydrology of the tributary area. This method of analysis has proven to be particularly effective in regional scale studies of groundwater conditions where direct methods such as the mapping of hydro-stratigraphic data are costly and time consuming to complete. In environments such as the Quaternary terrain of the Great Lakes basin, the prevailing opinion is that base flow is primarily the result of groundwater discharge to surface water features. Natural factors that approximate groundwater discharge such as the retention and delayed release of runoff by wetlands and lakes, and anthropogenic factors such as flow regulation and wastewater discharge, can be significant relative to groundwater discharge and must be identified prior to the characterization of groundwater conditions.

Various methods for the calculation of base flow are described in the literature. A frequently cited review of the topic that includes a detailed evaluation of two of these methods is presented by Natham and McMahon (1990). Many of the transformations that are applied to base flow data in the characterization of groundwater conditions require the aggregation and differentiation of the data. For example, daily data for a streamflow gauge can be totaled to derive monthly volumes of flow that can then be differentiated relative to values for tributary gauges in order to identify the contributions of the area that is immediately upstream of the gauge. Computations such as these can be conveniently performed in the context of a relational database management system or RDBMS and therefore the calculation of base flow in this same context facilitates subsequent analyses. The United Kingdom Institute of Hydrology or UKIH method for the calculation of base flow (Institute of Hydrology 1980) is one the methods that are evaluated by Natham and McMahon (1990) and can be expressed in terms of relatively simple RDBMS and related operations. The method has been successfully applied in many analyses including provincial scale studies for Ontario that are reported by Moin and Shaw (1985). These two features, the relative simplicity and robust performance of the UKIH method, are the basis for the selection of the method for this application. In summary, the method involves the division of input daily streamflow data into five-day segments, calculation of the minimum values of flow during each of these segments, comparison of the minimum values for each segment to the values for the adjacent segments, and selection of the minimum values as turning points relative to the adjacent values and using a quantitative constraint. Turning points are defined as observations of streamflow where the flow is assumed to be entirely base flow. The turning points are then interpolated to obtain a continuous time series that approximates the variation of the base flow with respect to time.

Summary of Relational Database Management System Functionality

RDBMS applications are increasingly common in office, enterprise, and Internet based computing settings; numerous commercial and public domain RDBMS software are now available; and a large volume of topical and software specific documentation has been published. Texts such as O'Neil (1994) provide detailed descriptions of RDBMS concepts that include the relatively simple functionality that is referenced in the following paragraphs and sections of this report.

RDBMS content is assembled into tables containing records or rows of data and fields or columns of the components of the data. In this application, streamflow data are initially assembled into the `flowbydate` table where the records of the table represent daily observations and the fields of the table are the gauge identifiers, years, months, and days of the observations (*station*, *year*, *month*, and *day*); observed flows (*flow*); and characteristics of the observations (*flag*). The UKIH method for the calculation of base flow is demonstrated using streamflow data for the Pang River at Pangborne for the period of January 1 to August 31, 1970 (Institute of Hydrology 1980). These data are listed in Table 1 using the structure of the `flowbydate` table. Content for the *flag* field is not provided in the demonstration data set and therefore this field is not included in Table 1. Three other tables are also used as input. The fields of the `dayno` table are the years, months, and days (*year*, *month*, and *day*) corresponding to a unique and monotonic day numbering scheme (*dayno*). In the case of the demonstration data set, *dayno* is calculated relative to January 1, 1970. The structure and content of the `dayno` table are shown in Table 2. The fields of the `segment` table are these same day numbers and a unique and monotonic numbering of the five-day segments of the data (*segment*) where the numbering of the segments is also calculated relative to January 1, 1970; that is, the first segment includes the data for January 1 to 5 (1 ≤ *dayno* ≤ 5), the second segment includes the data for January 6 to 10 (6 ≤ *dayno* ≤ 10), and so on. The implications of this arbitrary division of the data are assessed in a subsequent section of this report. The structure and content of the `segment` table are shown in Table 3. Finally, the fields of the adjsegment table are these same segment numbers and the numbers of the adjacent segments (*adjsegment*) with two records for each value of the *segment* field; specifically, with one record for the previous adjacent segment and one record for the subsequent adjacent segment. The structure and content of the *adjsegment* table are shown in Table 4. The data types of these fields are variously text (*station* and *flag*), integer (*year*, *month*, *day*, *dayno*, *segment*, and *adjsegment*), and floating point (*flow*). Results derived from the fields are typically of the same type as the source data.

Relations among the data are defined by joining, in a RDBMS sense, the fields of the tables and allow various transformations to be applied to the records. In this application, the *year*, *month*, and *day* fields of the `flowbydate` table are related to the *year*, *month*, and *day* fields of the `dayno` table and a query is used to construct the `flowbydayno` table where the fields of this new table are the gauge identifiers, day numbers, and flows (*station*, *dayno*, and *flow*) corresponding to each observation. The Structured Query Language or SQL is a standard syntax for the expression of RDBMS queries. The SQL syntax for the construction of the `flowbydayno` table and the first 25 records of output are listed in Table 5.

Groups of records are processed using aggregate functions. For example, the *station*, *year*, and *month* fields of the `flowbydate` table can each be grouped and the values of the *flow* field averaged over the groupings to derive average values of streamflow for each gauge, year, and month. Aggregate functions are used in numerous of the operations in this implementation (see, for example, Table 8).

Implementation of the Approach

The UKIH method for the calculation of base flow involves two tasks. The first of these tasks is the identification of turning points within input streamflow data; the second task is the interpolation of the resulting turning points. The steps that lead to the completion of these tasks are summarized in the following paragraphs using SQL and related code fragments. This implementation is based on UNIX shell programming (Arthur and Burns 1994), the Practical Extraction and Reporting Language or PERL (www.perl.org), the PostgreSQL RDBMS (www.postgresql.org), and the Octave language for numerical calculations (www.octave.org). All of these component software are public domain. Other publicly or commercially available software with matching functionality can be substituted for these components and verified using the demonstration input and output data that are cited in the following paragraphs.

Table 6 lists `basebydayno.pl`, a PERL code fragment that co-ordinates the calculation of base flow. A sequence of queries is completed using SQL code fragments input from `step1.sql` through `step6.sql` where `step1.sql` is derived from `step1.template` by the substitution of the current streamflow gauge identifier. The command line arguments for `basebydayno.pl` are the name of the database containing the *`flowbydayno`*, *`segment`*, and *`adjsegment`* tables and the name of the file into which the calculated values of base flow are output. This output can then either be input back into the same database or distributed for further use and analysis. The list of gauge identifiers for processing is taken from standard input and the steps in the calculation of base flow are as follows:

1. The first step extracts streamflow data for the current gauge and divides the data into the five-day segments required by the UKIH method. The query relates the *`dayno`* fields of the `flowbydayno` and `segment` tables and selects the *`segment`*, *`dayno`*, and *`flow`* fields from the matching records for the current value of the *`station`* field of the `flowbydayno` table. The selected records and fields are output to the `step1` table. The syntax of `step1.sql` and the first 25 records of output are listed in Table 7. These and the following selected records of output are based on the content of the `flowbydate`, `dayno`, `segment`, and `adjsegment` tables listed in Tables 1 through 4.

2. The second query determines the number of observations of streamflow within each five-day segment, and the minimum value of flow for each segment, by grouping the records of the `step1` table based on the *`segment`* field. Discontinuities in the input streamflow data may result in counts that are less than the expected value of five. The selected records and fields are output to the `step2` table. The syntax of `step2.sql` and the first 25 records of output are listed in Table 8.

3. The third step determines a candidate set of turning points. The query relates the *`segment`* and *`flow`* fields of the `step1` table to the *`segment`* and *`minofflow`* fields of the `step2` table and selects the

minimum values of the *dayno* and *flow* fields of the matching records for fully populated segments of data (i.e., segments with five records of data). The selected records and fields are output to the `step3` table. The syntax of `step3.sql` and the first 25 records of output are listed in Table 9.

4. The fourth step determines the minimum of the two minimum values of flow for the segments that are adjacent to each segment and counts the adjacent segments of results. Discontinuities in the input streamflow data may result in counts that are less than the expected value of two. The query relates the *adjsegment* field of the `adjsegment` table to the *segment* field of the `step3` table, counts the numbers of matching records using the *adjsegment* field, and determines the minimum values of the *minofflow* field. The selected records and fields are output to the `step4` table. The syntax of `step4.sql` and the first 25 records of output are listed in Table 10.

5. The fifth step determines the final set of turning points. The query relates the *segment* fields of the `step3` and `step4` tables and selects the *dayno* and *minofflow* fields from the matching records where minimum values of flow are available for both adjacent segments and where these values multiplied by 0.9 are less than the minimum of the two minimum values of flow for the adjacent segments. The syntax of `step5.sql` and the first 25 output records are listed in Table 11. These records are input by `basebydayno.pl` are then output to the `basebydayno.temp` file for subsequent processing.

6. The sixth and final query selects the *dayno* and *flow* fields from all of the records of the `step1` table and sorts the results into ascending order based on the *dayno* field. This step also removes from the database the temporary *step1*, *step2*, *step3*, and *step4* tables that were created during the previous queries in preparation for the calculation of base flow for another gauge. The syntax of `step6.sql` and the first 25 output records are listed in Table 12. These records are input by `basebydayno.pl` are then output to the `basebydayno.temp` file for subsequent processing.

7. The next task in the calculation of base flow is the interpolation of the final set of turning points. This task is completed using the Octave fragment `basebydayno.octave` that is listed in Table 13; the first 25 output records corresponding to the output listed in Tables 11 and 12 are also listed in Table 13. The `basebydayno.octave` fragment inputs the table of turning points (i.e., the *dayno* and *minofflow* fields of the records calculated using `step5.sql` and output by `basebydayno.pl`) and the table of total flow (i.e., the *dayno* and *flow* fields of the records calculated using `step6.sql` and output by `basebydayno.pl`) and then interpolates the table of turning points for each value of *dayno* in the table of total flow. Values of base flow are interpolated only for days that are within the calculated range of turning points and are not interpolated for days where data is missing due to a discontinuity in the observed values of total flow. This prevents the interpolation of base flow over

discontinuities in the input streamflow data using unrelated turning points located near the limits of the available data. The interpolated values of base flow are constrained such that the values are less than or equal to the corresponding values of total flow. This is a departure from the UKIH method and is assessed in the following section of this report. Interpolation is performed using a linear Lagrangian procedure (e.g., Gerald 1980) where the `lagrange.m` fragment that is listed in Table 14 is one example of the procedure (e.g., ftp.mathworks.com/pub/contrib/v4/approx).

The structure and content of the `basebydayno` table output by `basebydayno.pl` are shown in Table 15 and the observed values of total flow and calculated values of base flow are plotted in Figure 1. Division of the abscissa in Figure 1 indicates the five-day segments of data and the points shown along the traces of total and base flow indicate the calculated turning points. The final set of turning points identified using this procedure precisely matches the values cited in the documentation of the UKIH method. The five days for which the interpolated values of base flow are constrained using total flow are also indicated.

Arbitrary Aspects of the Implementation

This implementation of the UKIH method for the calculation of base flow constrains the interpolation of the turning points using the corresponding values of total flow. This is a logical but arbitrary departure from the standard method and results in differing calculated base flows. Division of the streamflow data into five-day segments is also arbitrary; for example, the division is performed relative to the first day of the input streamflow data in the case of the demonstration data set. This section of the report examines the implications of these two arbitrary aspects. The data that are used are for Water Survey of Canada streamflow gauge 02GA018, which is located on the Nith River at New Hamburg, and were extracted and reformatted from the HYDAT CD ROM (Environment Canada 1999). The gauge is within the watershed of the Grand River and is located approximately 75 km west of the western limit of Lake Ontario at Hamilton. The area that is tributary to the gauge is estimated to be 547 km^2 and the average annual streamflow recorded by the gauge and distributed over the tributary area is roughly 383 mm. The value of base flow index (the long-term average rate of base flow measured relative to total flow) determined for the gauge is 0.29 and is within the range of typical values for southern Ontario. The input streamflow data extends from January 1, 1970 to December 31, 1998 and includes 9,891 observations; 701 observations are missing during the period of June 14, 1989 to May 15, 1991.

In this analysis, the `dayno` table was calculated relative to January 1, 1900 and the `segment` and `adjsegment` tables were calculated accordingly. Base flow was calculated from the streamflow data using these reference conditions and the SQL and related code fragments that are listed Tables 6 through 14, and with two revisions that test the arbitrary aspects of the implementation. The first of these revisions is the removal of the constrain on the interpolation of the turning points that is applied in the tenth line of code in `basebydayno.octave`. Figure 2 illustrates the observed values of total flow and the calculated values of base flow with and without the constraint where the shaded portion of the plot indicates the accumulated discrepancy between the calculated flows during the period. The indicated period is March 4 to May 23, 1993 (34,031 ≤ *dayno* ≤ 34,111) and includes an abrupt change in streamflow that results in a large discrepancy between the reference and unconstrained base flows. The divisions of the abscissa of the plot indicate each five-day segment of data and the points shown along the traces of total and base flow indicate the turning points.

Constraining the calculated base flows using total flow decreases the flows relative to the standard method and therefore the discrepancy between the reference and unconstrained base flows is greater than or equal to zero. The discrepancy is zero on days when the interpolated base flow is less than the corresponding total flow and greater than zero otherwise. The reference values of base flow and the discrepancies between the reference and unconstrained values were calculated on a daily basis for the duration of the input streamflow data and averaged by year and then by both year and month over the 24 years with complete base flow data (i.e., 1971 through 1998 and 1992 through 1997). The results of this averaging are shown in Figure 3. Averaged by year, discrepancies for 11 years (46 percent of the values) are within 1 percent of the reference flows and all 24 of the discrepancies are within 10 percent. The median discrepancy is 1 percent of the reference flows. Averaged by year and month, discrepancies for 247 months (86 percent of the values) are within 1 percent of the reference flows and 274 (95 percent) are within 10 percent. The median discrepancy is 0.04 percent of the reference flows and is less than the previous value due to the large number of months with a discrepancy of zero. Thus, the implications of constraining the data are generally modest at both the yearly and monthly scales, although significant discrepancies do occasionally occur for relatively low values of monthly base flow. The discrepancies are greater than or equal to zero and therefore the base flows calculated using this implementation systematically underestimate the flows that would be calculated using a strict implementation of the UKIH method. The magnitude of this departure is unlikely to be greater than a few percent of the calculated flows when measured on a long-term basis.

Division of the streamflow data into five-day segments is also arbitrary and is a function of first day to which a segment number is assigned. To test the implications of the division of the data, the segment numbering scheme was displaced forward by one through four days relative to the reference condition, resulting in four new versions of the `segment` table. When displaced by multiples of five days, the numbering of the segments is changed uniformly relative to the reference condition and there is no change in the calculated base flows. Figure 4 illustrates the implications of these calculations over the period used in the previous analysis. Displacing the segments by one day resulted in a very modest change in the calculated base flows. In contrast, displacing the segments by two through four days resulted in a substantial change in the calculated base flows where, again, the indicated period is characterized by highly variable streamflow and therefore is particularly sensitive to details of the calculation of base flow. The shaded portion of the plot indicates the accumulated discrepancy between the minimum and maximum of the five sets of daily values of base flow. The average of the five sets is also shown in Figure 4. This average is independent of the division of the streamflow data into five-day segments because all five unique divisions are reflected in the value. Calculation of base flow using five successive divisions of the input streamflow data where each division is displaced by one day, followed by averaging of the results, may therefore be more appropriate than using a single division of the data.

The reference values of base flow and the discrepancies between the reference values and the values calculated for one through four day displacements of the segment numbering scheme were calculated on a daily basis and averaged by year and then by year and month. The results of this averaging are plotted in Figure 5. The averaged discrepancies are both positive and negative where a positive value indicates that the value for the displaced segment numbering scheme is greater than the reference value. Averaged by year, discrepancies for 24 of the $4 \times 24 = 96$ years of data (25 percent of the values) are within ±1 percent of the reference flows and 68 (71 percent) are within ±10 percent. The median discrepancy is 3 percent of the reference flows. Averaged by year and month, discrepancies for 509 of the $4 \times 24 \times 12 = 1152$ months of data (44 percent of the values) are within ±1 percent of the reference flows and 997 (87 percent) are within ±10 percent. The median discrepancy is 1 percent of the reference flows and is less than the previous value due to the large number of months with a discrepancy of zero. Several discrepancies are greater than 100 percent of the reference flows where, similar to the previous analysis, these large discrepancies occur more frequently for relatively low values of monthly base flow.

Observations and Conclusions

Implementation of the UKIH method for the calculation of base flow using the approach that is described in this report enables the calculation to be performed in a RDBMS context. Because streamflow monitoring data are often managed and analyzed in this context, this approach may be more convenient than implementations that perform the calculation of base flow outside of the context. In addition, the SQL, PERL, and Octave code fragments that implement the approach and are listed in this report are likely to be portable to other component software and operating systems. Thus, the RDBMS compatibility and portability of this implementation may enable the application of the approach in a range of settings.

Two arbitrary aspects of the implementation, constraining the interpolation of the turning points such that base flow does exceed total flow and division of the streamflow data into the five-day segments that are required by the UKIH method, both influence the calculated base flows. Constraining the interpolation results in flows that are systematically less than the corresponding flows calculated using a strict implementation of the method. In most cases, the discrepancies between the flows are on the order of

1 percent of the flows and are unlikely to significantly influence subsequent analyses. Displacement of the division of the data into five-day segments relative to the reference division by one through four days results in discrepancies that, in most cases, are on the order of a few percent of the flows and are also unlikely to significantly influence subsequent analyses. Averaging the results of the five calculations eliminates the dependence of the calculated base flows on the segment numbering scheme and can be readily implemented in the RDBMS context of the calculations.

References

Arthur, L.J., and Burns, T. 1994. UNIX shell programming. Third edition. John Wiley and Sons, New York, New York.

Environment Canada. 1999. HYDAT Version 98-1.05.8. Surface water and sediment data to 1998. Environment Canada, Toronto, Ontario, Canada.

Gerald, C.F. 1980. Applied numerical analysis. Second edition. Addison-Wesley Publishing Company, Reading, Massachusetts.

Institute of Hydrology. 1980. Low flow studies, research reports 1 and 3. Institute of Hydrology, Wallingford, Oxon, UK.

Moin, S.M.A., and Shaw, M.A. 1985. Regional flood frequency analysis for Ontario streams, volumes 1-3. Environment Canada and Ontario Ministry of Natural Resources, Toronto, Ontario, Canada.

Nathan, R.J., and McMahon, T.A. 1990. Evaluation of automated techniques for base flow and recession analyses. Water Resources Research, 26(7): 1465-1473.

O'Neil, P. 1994. Database principles, programming, performance. Morgan Kaufmann Publishers, San Francisco, California.

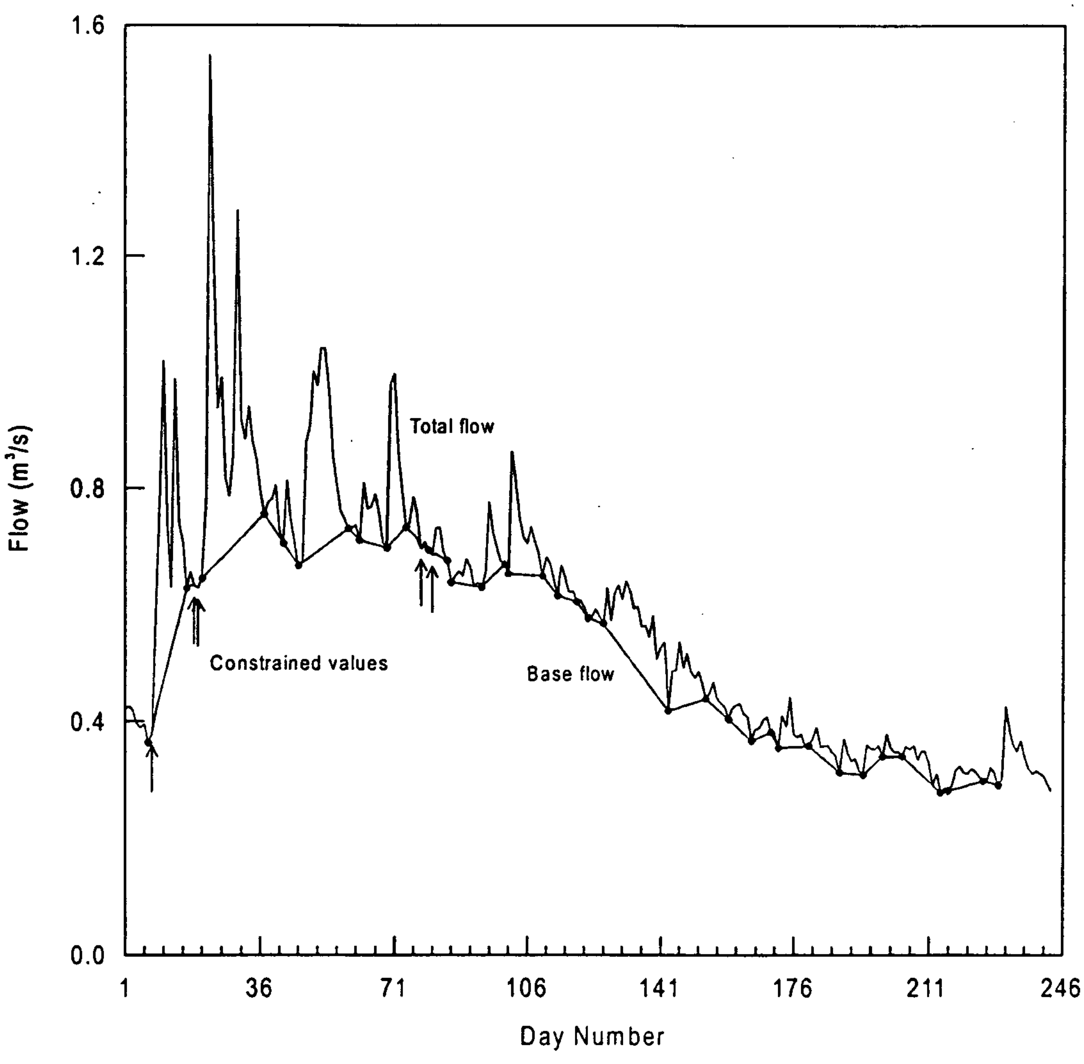


Figure 1. Results of the calculation of base flow using the demonstration data set.

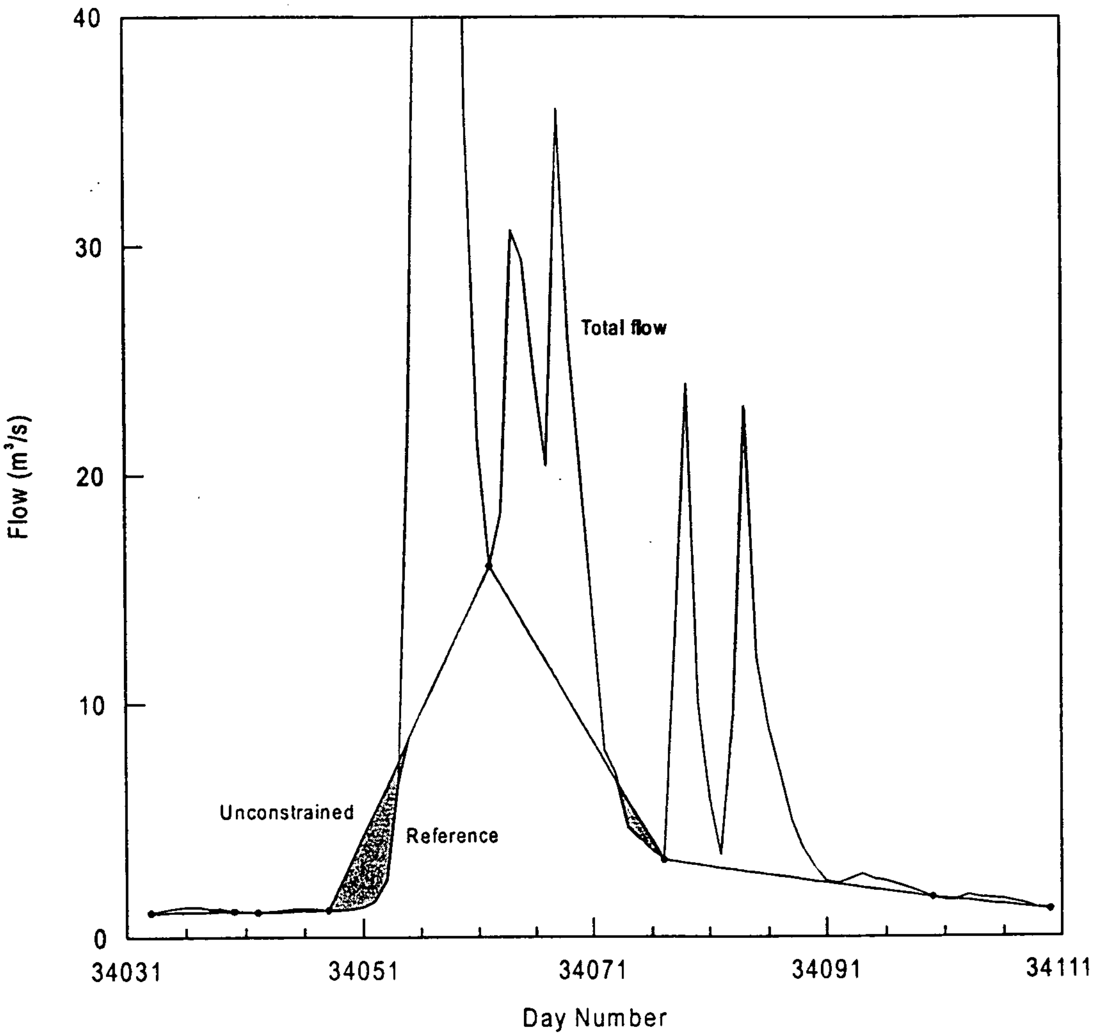


Figure 2. Comparison of the reference and unconstrained base flows.

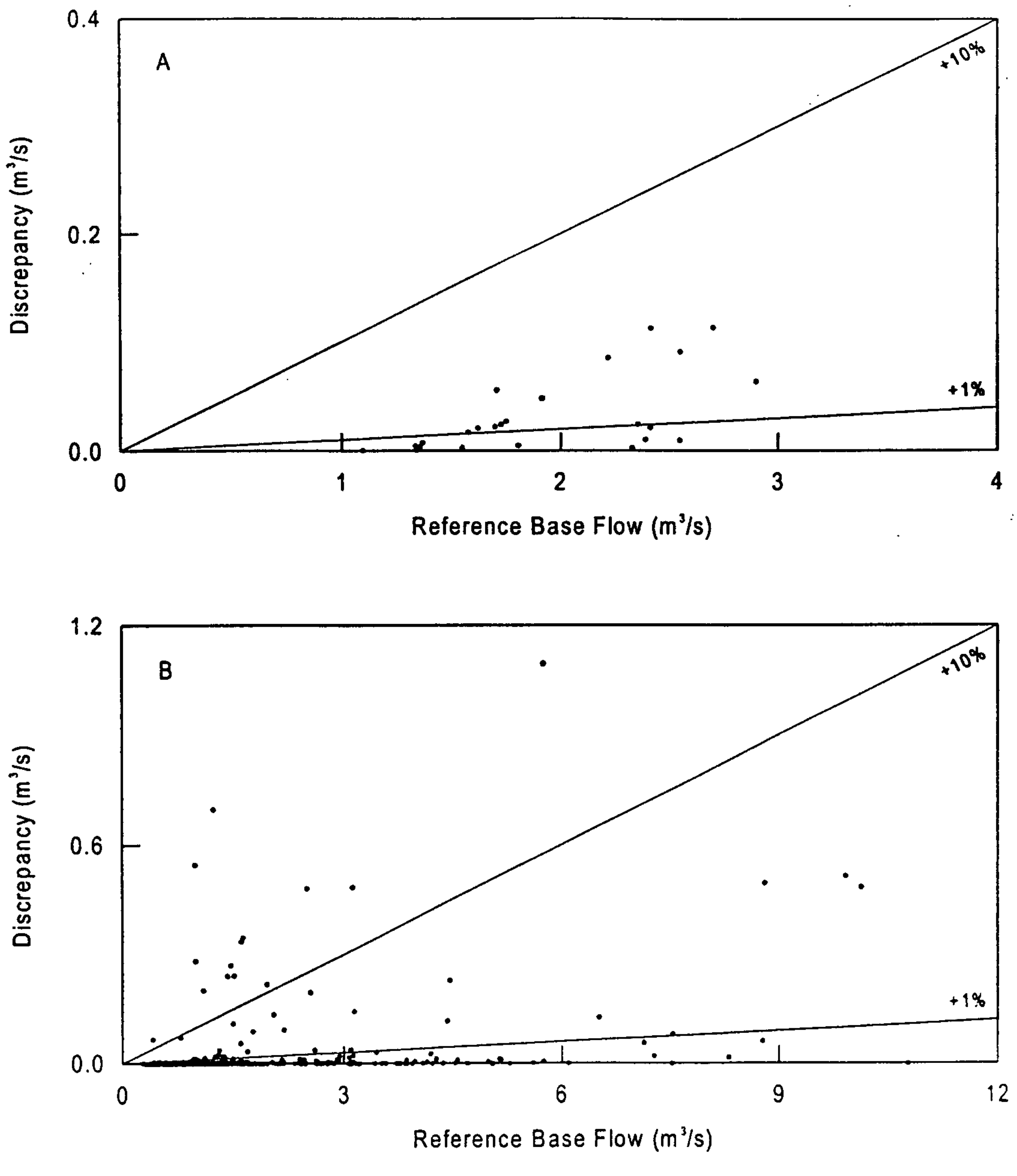


Figure 3. Yearly (A) and monthly (B) average discrepancies between the reference and unconstrained base flows.

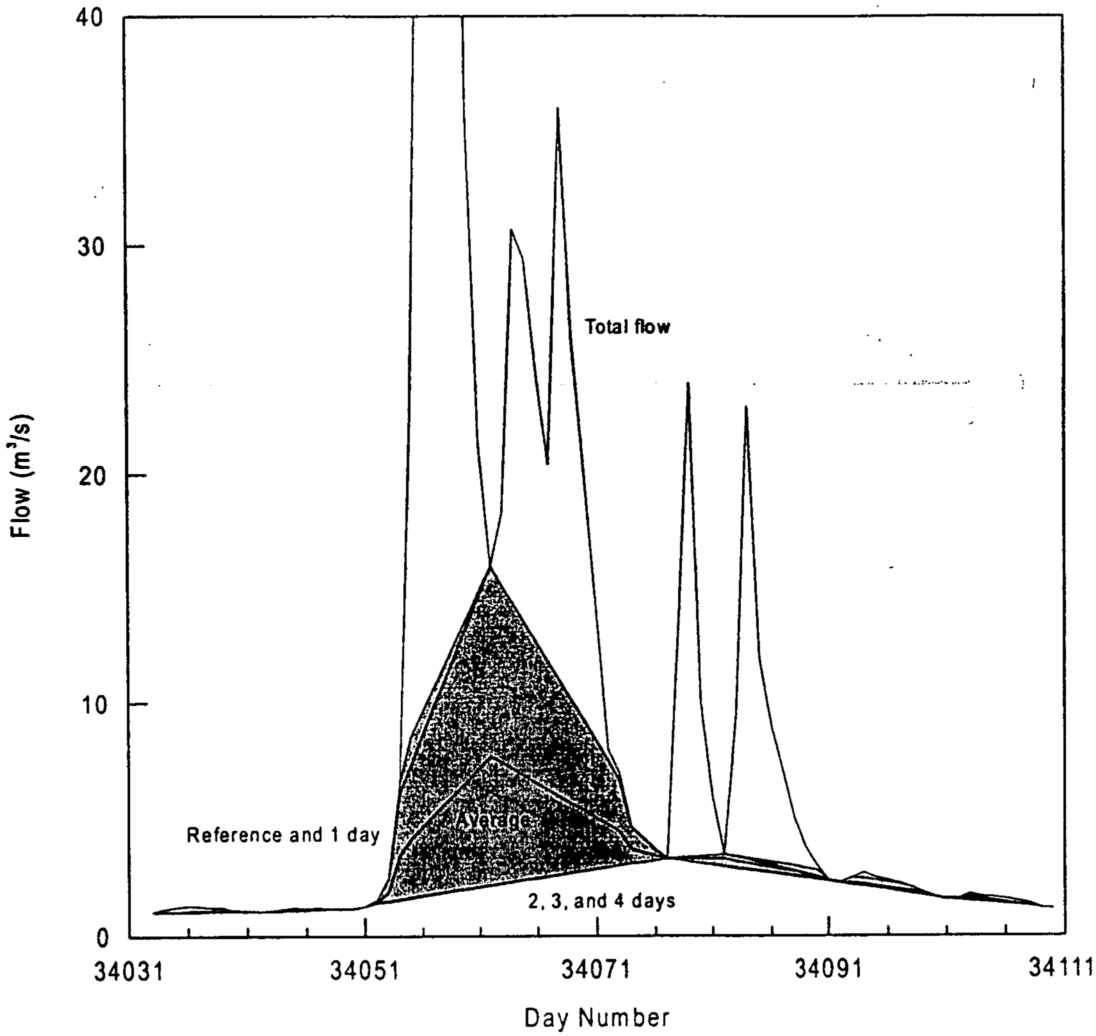
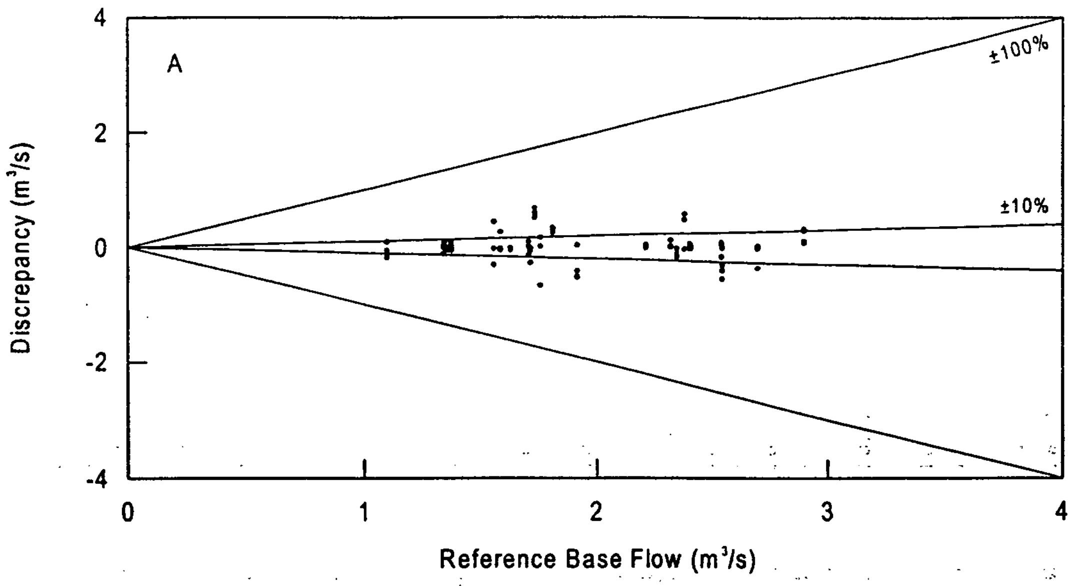


Figure 4. Comparison of the reference base flows and the flows calculated for displaced versions of the segment numbering scheme.



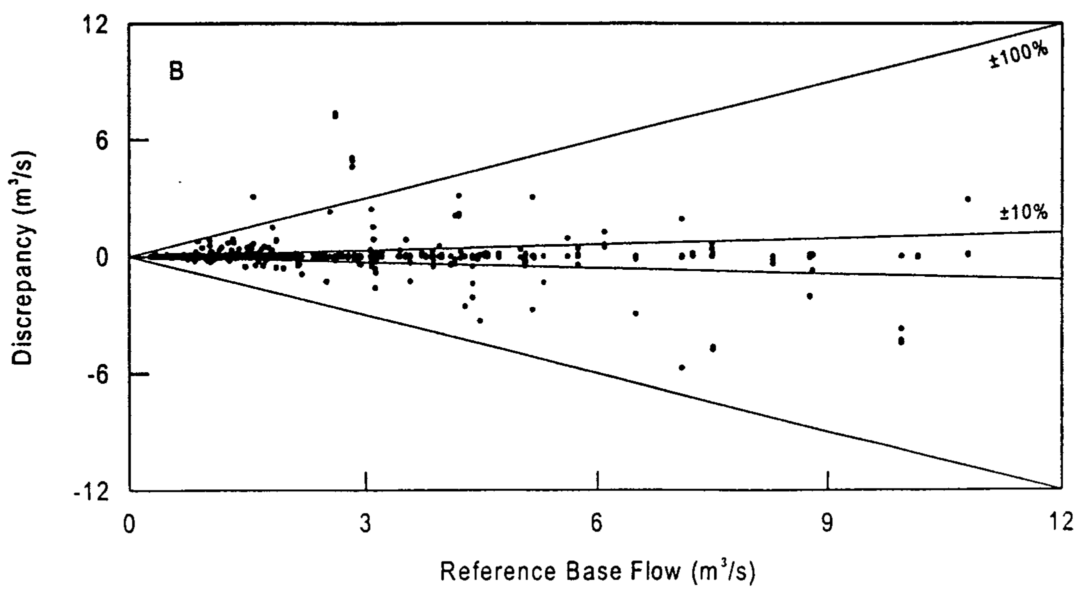


Figure 5. Yearly (A) and monthly (B) average discrepancies between the reference base flows and the flows calculated for displaced versions of the segment numbering scheme.

Table 1. Structure and content of the `flowbydate` table.

station	*year*	*month*	*day*	*flow*
PANG	1970	1	1	0.422
PANG	1970	1	2	0.426
PANG	1970	1	3	0.421
PANG	1970	1	4	0.398
PANG	1970	1	5	0.389
PANG	1970	1	6	0.395
PANG	1970	1	7	0.364
PANG	1970	1	8	0.379
PANG	1970	1	9	0.593
PANG	1970	1	10	0.775
PANG	1970	1	11	1.02
PANG	1970	1	12	0.74
PANG	1970	1	13	0.63
PANG	1970	1	14	0.988
PANG	1970	1	15	0.74
PANG	1970	1	16	0.708
PANG	1970	1	17	0.628
PANG	1970	1	18	0.657
PANG	1970	1	19	0.633
PANG	1970	1	20	0.628
PANG	1970	1	21	0.645
PANG	1970	1	22	0.784
PANG	1970	1	23	1.55
PANG	1970	1	24	1.18
PANG	1970	1	25	0.937
PANG	1970	1	26	0.99
PANG	1970	1	27	0.82
PANG	1970	1	28	0.786
PANG	1970	1	29	0.856
PANG	1970	1	30	1.28
PANG	1970	1	31	0.916
PANG	1970	2	1	0.883
PANG	1970	2	2	0.941
PANG	1970	2	3	0.879
PANG	1970	2	4	0.851

station	*year*	*month*	*day*	*flow*
PANG	1970	2	5	0.797
PANG	1970	2	6	0.756
PANG	1970	2	7	0.776
PANG	1970	2	8	0.781
PANG	1970	2	9	0.805
PANG	1970	2	10	0.729
PANG	1970	2	11	0.706
PANG	1970	2	12	0.813
PANG	1970	2	13	0.749
PANG	1970	2	14	0.708
PANG	1970	2	15	0.667
PANG	1970	2	16	0.673
PANG	1970	2	17	0.879
PANG	1970	2	18	0.907
PANG	1970	2	19	1
PANG	1970	2	20	0.975
PANG	1970	2	21	1.04
PANG	1970	2	22	1.04
PANG	1970	2	23	0.962
PANG	1970	2	24	0.85
PANG	1970	2	25	0.806
PANG	1970	2	26	0.761
PANG	1970	2	27	0.748
PANG	1970	2	28	0.731
PANG	1970	3	1	0.733
PANG	1970	3	2	0.737
PANG	1970	3	3	0.711
PANG	1970	3	4	0.81
PANG	1970	3	5	0.765
PANG	1970	3	6	0.768
PANG	1970	3	7	0.79
PANG	1970	3	8	0.759
PANG	1970	3	9	0.708
PANG	1970	3	10	0.698
PANG	1970	3	11	0.978

Table 1. Structure and content of the `flowbydate` table (continued).

station	*year*	*month*	*day*	*flow*	*station*	*year*	*month*	*day*	*flow*
PANG	1970	3	12	0.997	PANG	1970	4	16	0.706
PANG	1970	3	13	0.859	PANG	1970	4	17	0.735
PANG	1970	3	14	0.798	PANG	1970	4	18	0.71
PANG	1970	3	15	0.734	PANG	1970	4	19	0.689
PANG	1970	3	16	0.741	PANG	1970	4	20	0.649
PANG	1970	3	17	0.785	PANG	1970	4	21	0.682
PANG	1970	3	18	0.759	PANG	1970	4	22	0.672
PANG	1970	3	19	0.697	PANG	1970	4	23	0.644
PANG	1970	3	20	0.708	PANG	1970	4	24	0.615
PANG	1970	3	21	0.694	PANG	1970	4	25	0.669
PANG	1970	3	22	0.686	PANG	1970	4	26	0.647
PANG	1970	3	23	0.732	PANG	1970	4	27	0.622
PANG	1970	3	24	0.734	PANG	1970	4	28	0.622
PANG	1970	3	25	0.692	PANG	1970	4	29	0.605
PANG	1970	3	26	0.676	PANG	1970	4	30	0.607
PANG	1970	3	27	0.638	PANG	1970	5	1	0.596
PANG	1970	3	28	0.648	PANG	1970	5	2	0.577
PANG	1970	3	29	0.658	PANG	1970	5	3	0.58
PANG	1970	3	30	0.649	PANG	1970	5	4	0.592
PANG	1970	3	31	0.679	PANG	1970	5	5	0.58
PANG	1970	4	1	0.665	PANG	1970	5	6	0.568
PANG	1970	4	2	0.633	PANG	1970	5	7	0.628
PANG	1970	4	3	0.637	PANG	1970	5	8	0.572
PANG	1970	4	4	0.63	PANG	1970	5	9	0.619
PANG	1970	4	5	0.657	PANG	1970	5	10	0.634
PANG	1970	4	6	0.777	PANG	1970	5	11	0.608
PANG	1970	4	7	0.724	PANG	1970	5	12	0.641
PANG	1970	4	8	0.699	PANG	1970	5	13	0.624
PANG	1970	4	9	0.675	PANG	1970	5	14	0.593
PANG	1970	4	10	0.669	PANG	1970	5	15	0.596
PANG	1970	4	11	0.653	PANG	1970	5	16	0.563
PANG	1970	4	12	0.863	PANG	1970	5	17	0.564
PANG	1970	4	13	0.806	PANG	1970	5	18	0.545
PANG	1970	4	14	0.748	PANG	1970	5	19	0.581
PANG	1970	4	15	0.715	PANG	1970	5	20	0.506

Table 1. Structure and content of the `flowbydate` table (continued).

station	*year*	*month*	*day*	*flow*
PANG	1970	5	21	0.526
PANG	1970	5	22	0.536
PANG	1970	5	23	0.418
PANG	1970	5	24	0.487
PANG	1970	5	25	0.488
PANG	1970	5	26	0.538
PANG	1970	5	27	0.491
PANG	1970	5	28	0.517
PANG	1970	5	29	0.486
PANG	1970	5	30	0.475
PANG	1970	5	31	0.485
PANG	1970	6	1	0.466
PANG	1970	6	2	0.439
PANG	1970	6	3	0.449
PANG	1970	6	4	0.468
PANG	1970	6	5	0.44
PANG	1970	6	6	0.431
PANG	1970	6	7	0.426
PANG	1970	6	8	0.404
PANG	1970	6	9	0.424
PANG	1970	6	10	0.428
PANG	1970	6	11	0.431
PANG	1970	6	12	0.414
PANG	1970	6	13	0.407
PANG	1970	6	14	0.367
PANG	1970	6	15	0.387
PANG	1970	6	16	0.389
PANG	1970	6	17	0.403
PANG	1970	6	18	0.408
PANG	1970	6	19	0.382
PANG	1970	6	20	0.378
PANG	1970	6	21	0.355
PANG	1970	6	22	0.411
PANG	1970	6	23	0.391
PANG	1970	6	24	0.443
PANG	1970	6	25	0.379
PANG	1970	6	26	0.373
PANG	1970	6	27	0.378
PANG	1970	6	28	0.359
PANG	1970	6	29	0.358
PANG	1970	6	30	0.37
PANG	1970	7	1	0.39
PANG	1970	7	2	0.357
PANG	1970	7	3	0.358
PANG	1970	7	4	0.359
PANG	1970	7	5	0.347
PANG	1970	7	6	0.341
PANG	1970	7	7	0.312
PANG	1970	7	8	0.371
PANG	1970	7	9	0.347
PANG	1970	7	10	0.332
PANG	1970	7	11	0.335
PANG	1970	7	12	0.316
PANG	1970	7	13	0.308
PANG	1970	7	14	0.359
PANG	1970	7	15	0.354
PANG	1970	7	16	0.352
PANG	1970	7	17	0.359
PANG	1970	7	18	0.34
PANG	1970	7	19	0.379
PANG	1970	7	20	0.355
PANG	1970	7	21	0.347
PANG	1970	7	22	0.348
PANG	1970	7	23	0.34
PANG	1970	7	24	0.357
PANG	1970	7	25	0.352
PANG	1970	7	26	0.358
PANG	1970	7	27	0.333
PANG	1970	7	28	0.349
PANG	1970	7	29	0.348

Table 1. Structure and content of the `flowbydate` table (continued).

station	*year*	*month*	*day*	*flow*
PANG	1970	7	30	0.328
PANG	1970	7	31	0.291
PANG	1970	8	1	0.31
PANG	1970	8	2	0.278
PANG	1970	8	3	0.286
PANG	1970	8	4	0.281
PANG	1970	8	5	0.29
PANG	1970	8	6	0.317
PANG	1970	8	7	0.323
PANG	1970	8	8	0.313
PANG	1970	8	9	0.31
PANG	1970	8	10	0.319
PANG	1970	8	11	0.315
PANG	1970	8	12	0.305
PANG	1970	8	13	0.298
PANG	1970	8	14	0.299
PANG	1970	8	15	0.32
PANG	1970	8	16	0.313
PANG	1970	8	17	0.291
PANG	1970	8	18	0.301
PANG	1970	8	19	0.427
PANG	1970	8	20	0.384
PANG	1970	8	21	0.359
PANG	1970	8	22	0.349
PANG	1970	8	23	0.368
PANG	1970	8	24	0.34
PANG	1970	8	25	0.32
PANG	1970	8	26	0.31
PANG	1970	8	27	0.315
PANG	1970	8	28	0.311
PANG	1970	8	29	0.306
PANG	1970	8	30	0.293
PANG	1970	8	31	0.28

Table 2. Structure and content of the dayno table.

year	*month*	*day*	*dayno*
1970	1	1	1
1970	1	2	2
1970	1	3	3
1970	1	4	4
1970	1	5	5
1970	1	6	6
1970	1	7	7
1970	1	8	8
1970	1	9	9
1970	1	10	10
1970	1	11	11
1970	1	12	12
1970	1	13	13
1970	1	14	14
1970	1	15	15
1970	1	16	16
1970	1	17	17
1970	1	18	18
1970	1	19	19
1970	1	20	20
1970	1	21	21
1970	1	22	22
1970	1	23	23
1970	1	24	24
1970	1	25	25
1970	1	26	26
1970	1	27	27
1970	1	28	28
1970	1	29	29
1970	1	30	30
1970	1	31	31
1970	2	1	32
1970	2	2	33
1970	2	3	34
1970	2	4	35
1970	2	5	36
1970	2	6	37
1970	2	7	38
1970	2	8	39
1970	2	9	40
1970	2	10	41
1970	2	11	42
1970	2	12	43
1970	2	13	44
1970	2	14	45
1970	2	15	46
1970	2	16	47
1970	2	17	48
1970	2	18	49
1970	2	19	50
1970	2	20	51
1970	2	21	52
1970	2	22	53
1970	2	23	54
1970	2	24	55
1970	2	25	56
1970	2	26	57
1970	2	27	58
1970	2	28	59
1970	3	1	60
1970	3	2	61
1970	3	3	62
1970	3	4	63
1970	3	5	64
1970	3	6	65
1970	3	7	66
1970	3	8	67
1970	3	9	68
1970	3	10	69
1970	3	11	70

Table 2. Structure and content of the dayno table (continued).

year	*month*	*day*	*dayno*
1970	3	12	71
1970	3	13	72
1970	3	14	73
1970	3	15	74
1970	3	16	75
1970	3	17	76
1970	3	18	77
1970	3	19	78
1970	3	20	79
1970	3	21	80
1970	3	22	81
1970	3	23	82
1970	3	24	83
1970	3	25	84
1970	3	26	85
1970	3	27	86
1970	3	28	87
1970	3	29	88
1970	3	30	89
1970	3	31	90
1970	4	1	91
1970	4	2	92
1970	4	3	93
1970	4	4	94
1970	4	5	95
1970	4	6	96
1970	4	7	97
1970	4	8	98
1970	4	9	99
1970	4	10	100
1970	4	11	101
1970	4	12	102
1970	4	13	103
1970	4	14	104
1970	4	15	105
1970	4	16	106
1970	4	17	107
1970	4	18	108
1970	4	19	109
1970	4	20	110
1970	4	21	111
1970	4	22	112
1970	4	23	113
1970	4	24	114
1970	4	25	115
1970	4	26	116
1970	4	27	117
1970	4	28	118
1970	4	29	119
1970	4	30	120
1970	5	1	121
1970	5	2	122
1970	5	3	123
1970	5	4	124
1970	5	5	125
1970	5	6	126
1970	5	7	127
1970	5	8	128
1970	5	9	129
1970	5	10	130
1970	5	11	131
1970	5	12	132
1970	5	13	133
1970	5	14	134
1970	5	15	135
1970	5	16	136
1970	5	17	137
1970	5	18	138
1970	5	19	139
1970	5	20	140

Table 2. Structure and content of the dayno table (continued).

year	*month*	*day*	*dayno*
1970	5	21	141
1970	5	22	142
1970	5	23	143
1970	5	24	144
1970	5	25	145
1970	5	26	146
1970	5	27	147
1970	5	28	148
1970	5	29	149
1970	5	30	150
1970	5	31	151
1970	6	1	152
1970	6	2	153
1970	6	3	154
1970	6	4	155
1970	6	5	156
1970	6	6	157
1970	6	7	158
1970	6	8	159
1970	6	9	160
1970	6	10	161
1970	6	11	162
1970	6	12	163
1970	6	13	164
1970	6	14	165
1970	6	15	166
1970	6	16	167
1970	6	17	168
1970	6	18	169
1970	6	19	170
1970	6	20	171
1970	6	21	172
1970	6	22	173
1970	6	23	174
1970	6	24	175
1970	6	25	176
1970	6	26	177
1970	6	27	178
1970	6	28	179
1970	6	29	180
1970	6	30	181
1970	7	1	182
1970	7	2	183
1970	7	3	184
1970	7	4	185
1970	7	5	186
1970	7	6	187
1970	7	7	188
1970	7	8	189
1970	7	9	190
1970	7	10	191
1970	7	11	192
1970	7	12	193
1970	7	13	194
1970	7	14	195
1970	7	15	196
1970	7	16	197
1970	7	17	198
1970	7	18	199
1970	7	19	200
1970	7	20	201
1970	7	21	202
1970	7	22	203
1970	7	23	204
1970	7	24	205
1970	7	25	206
1970	7	26	207
1970	7	27	208
1970	7	28	209
1970	7	29	210

Table 2. Structure and content of the `dayno` table (continued).

year	*month*	*day*	*dayno*
1970	7	30	211
1970	7	31	212
1970	8	1	213
1970	8	2	214
1970	8	3	215
1970	8	4	216
1970	8	5	217
1970	8	6	218
1970	8	7	219
1970	8	8	220
1970	8	9	221
1970	8	10	222
1970	8	11	223
1970	8	12	224
1970	8	13	225
1970	8	14	226
1970	8	15	227
1970	8	16	228
1970	8	17	229
1970	8	18	230
1970	8	19	231
1970	8	20	232
1970	8	21	233
1970	8	22	234
1970	8	23	235
1970	8	24	236
1970	8	25	237
1970	8	26	238
1970	8	27	239
1970	8	28	240
1970	8	29	241
1970	8	30	242
1970	8	31	243

Table 3. Structure and content of the `segment` table.

dayno	*segment*	*dayno*	*segment*	*dayno*	*segment*	*dayno*	*segment*
1	1	36	8	71	15	106	22
2	1	37	8	72	15	107	22
3	1	38	8	73	15	108	22
4	1	39	8	74	15	109	22
5	1	40	8	75	15	110	22
6	2	41	9	76	16	111	23
7	2	42	9	77	16	112	23
8	2	43	9	78	16	113	23
9	2	44	9	79	16	114	23
10	2	45	9	80	16	115	23
11	3	46	10	81	17	116	24
12	3	47	10	82	17	117	24
13	3	48	10	83	17	118	24
14	3	49	10	84	17	119	24
15	3	50	10	85	17	120	24
16	4	51	11	86	18	121	25
17	4	52	11	87	18	122	25
18	4	53	11	88	18	123	25
19	4	54	11	89	18	124	25
20	4	55	11	90	18	125	25
21	5	56	12	91	19	126	26
22	5	57	12	92	19	127	26
23	5	58	12	93	19	128	26
24	5	59	12	94	19	129	26
25	5	60	12	95	19	130	26
26	6	61	13	96	20	131	27
27	6	62	13	97	20	132	27
28	6	63	13	98	20	133	27
29	6	64	13	99	20	134	27
30	6	65	13	100	20	135	27
31	7	66	14	101	21	136	28
32	7	67	14	102	21	137	28
33	7	68	14	103	21	138	28
34	7	69	14	104	21	139	28
35	7	70	14	105	21	140	28

Table 3. Structure and content of the `segment` table (continued).

dayno	*segment*	*dayno*	*segment*	*dayno*	*segment*
141	29	176	36	211	43
142	29	177	36	212	43
143	29	178	36	213	43
144	29	179	36	214	43
145	29	180	36	215	43
146	30	181	37	216	44
147	30	182	37	217	44
148	30	183	37	218	44
149	30	184	37	219	44
150	30	185	37	220	44
151	31	186	38	221	45
152	31	187	38	222	45
153	31	188	38	223	45
154	31	189	38	224	45
155	31	190	38	225	45
156	32	191	39	226	46
157	32	192	39	227	46
158	32	193	39	228	46
159	32	194	39	229	46
160	32	195	39	230	46
161	33	196	40	231	47
162	33	197	40	232	47
163	33	198	40	233	47
164	33	199	40	234	47
165	33	200	40	235	47
166	34	201	41	236	48
167	34	202	41	237	48
168	34	203	41	238	48
169	34	204	41	239	48
170	34	205	41	240	48
171	35	206	42	241	49
172	35	207	42	242	49
173	35	208	42	243	49
174	35	209	42		
175	35	210	42		

Table 4. Structure and content of the `adjsegment` table.

segment	*adjsegment*
1	0
1	2
2	1
2	3
3	2
3	4
4	3
4	5
5	4
5	6
6	5
6	7
7	6
7	8
8	7
8	9
9	8
9	10
10	9
10	11
11	10
11	12
12	11
12	13
13	12
13	14
14	13
14	15
15	14
15	16
16	15
16	17
17	16
17	18
18	17
18	19
19	18
19	20
20	19
20	21
21	20
21	22
22	21
22	23
23	22
23	24
24	23
24	25
25	24
25	26
26	25
26	27
27	26
27	28
28	27
28	29
29	28
29	30
30	29
30	31
31	30
31	32
32	31
32	33
33	32
33	34
34	33
34	35
35	34
35	36
36	35
36	37
37	36
37	38
38	37
38	39
39	38
39	40
40	39
40	41
41	40
41	42
42	41
42	43
43	42
43	44
44	43
44	45
45	44
45	46
46	45
46	47
47	46
47	48
48	47
48	49
49	48
49	50

Table 5. Syntax of `flowbydayno.sql` and selected records of output.

```
select       flowbydate.station,
             dayno.dayno,
             flowbydate.flow
into table flowbydayno
from         dayno,
             flowbydate
where        dayno.year=flowbydate.year and
             dayno.month=flowbydate.month and
             dayno.day=flowbydate.day ;
```

station	*dayno*	*flow*
PANG	1	0.422
PANG	2	0.426
PANG	3	0.421
PANG	4	0.398
PANG	5	0.389
PANG	6	0.395
PANG	7	0.364
PANG	8	0.379
PANG	9	0.593
PANG	10	0.775
PANG	11	1.02
PANG	12	0.74
PANG	13	0.63
PANG	14	0.988
PANG	15	0.74
PANG	16	0.708
PANG	17	0.628
PANG	18	0.657
PANG	19	0.633
PANG	20	0.628
PANG	21	0.645
PANG	22	0.784
PANG	23	1.55
PANG	24	1.18
PANG	25	0.937

Table 6. Syntax of basebydayno.pl.

```
#!/usr/bin/perl

($database,$outfile) = @ARGV ;

while(<STDIN>) {
  chop ;
  $station = $_ ;
  system("sed -e 's/STATION/$station/g' step1.template > step1.sql") ;
  system("psql -d $database -q -f step1.sql") ;
  system("psql -d $database -q -f step2.sql") ;
  system("psql -d $database -q -f step3.sql") ;
  system("psql -d $database -q -f step4.sql") ;
  $_ =   `psql -d $database -q -f step5.sql -A -F ' ' -t` ;
  chop ;
  $n1 = tr/\n// + 1 ;
  $header = "# name: table1\n# type: matrix\n# rows: $n1\n# columns: 2" ;
  system("echo '$header\n$_' >  basebydayno.temp") ;
  $_ =   `psql -d $database -q -f step6.sql -A -F ' ' -t` ;
  chop ;
  $n2 = tr/\n// + 1 ;
  $header = "# name: table2\n# type: matrix\n# rows: $n2\n# columns: 2" ;
  system("echo '$header\n$_' >> basebydayno.temp") ;
  system("octave -q -f basebydayno.octave |
          grep '.' |
          sed -e 's/^/$station /' |
          tr -s ' ' |
          tr ' ' '\t' >> $outfile") ;
  system("rm step1.sql basebydayno.temp") ;
}
```

Table 7. Syntax of `step1.sql` and selected records of output.

```
select      segment.segment,
            flowbydayno.dayno,
            flowbydayno.flow
into table  step1
from        flowbydayno,
            segment
where       flowbydayno.dayno=segment.dayno and
            flowbydayno.station='PANG' ;
```

segment	*dayno*	*flow*
1	1	0.422
1	2	0.426
1	3	0.421
1	4	0.398
1	5	0.389
2	6	0.395
2	7	0.364
2	8	0.379
2	9	0.593
2	10	0.775
3	11	1.02
3	12	0.74
3	13	0.63
3	14	0.988
3	15	0.74
4	16	0.708
4	17	0.628
4	18	0.657
4	19	0.633
4	20	0.628
5	21	0.645
5	22	0.784
5	23	1.55
5	24	1.18
5	25	0.937

Table 8. Syntax of `step2.sql` and selected records of output.

```
select      step1.segment,
            count(step1.flow) as countofflow,
            min(step1.flow) as minofflow
into table  step2
from        step1
group by    step1.segment ;
```

segment	*countofflow*	*minofflow*
1	5	0.389
2	5	0.364
3	5	0.63
4	5	0.628
5	5	0.645
6	5	0.786
7	5	0.851
8	5	0.756
9	5	0.706
10	5	0.667
11	5	0.85
12	5	0.731
13	5	0.711
14	5	0.698
15	5	0.734
16	5	0.694
17	5	0.676
18	5	0.638
19	5	0.63
20	5	0.669
21	5	0.653
22	5	0.649
23	5	0.615
24	5	0.605
25	5	0.577

Table 9. Syntax of `step3.sql` and selected records of output.

```
select      step2.segment,
            min(step1.dayno) as dayno,
            min(step1.flow) as minofflow
into table  step3
from        step1,
            step2
where       step1.segment=step2.segment and
            step1.flow=step2.minofflow and
            step2.countofflow=5
group by    step2.segment ;
```

segment	*dayno*	*minofflow*
1	5	0.389
2	7	0.364
3	13	0.63
4	17	0.628
5	21	0.645
6	28	0.786
7	35	0.851
8	37	0.756
9	42	0.706
10	46	0.667
11	55	0.85
12	59	0.731
13	62	0.711
14	69	0.698
15	74	0.734
16	80	0.694
17	85	0.676
18	86	0.638
19	94	0.63
20	100	0.669
21	101	0.653
22	110	0.649
23	114	0.615
24	119	0.605
25	122	0.577

Table 10. Syntax of `step4.sql` and selected records of output.

```
select        adjsegment.segment,
              count(adjsegment.adjsegment) as countofadjsegment,
              min(step3.minofflow) as minofminofflow
into table step4
from          adjsegment,
              step3
where         adjsegment.adjsegment=step3.segment
group by      adjsegment.segment ;
```

segment	*countofadjsegment*	*minofminofflow*
1	1	0.364
2	2	0.389
3	2	0.364
4	2	0.63
5	2	0.628
6	2	0.645
7	2	0.756
8	2	0.706
9	2	0.667
10	2	0.706
11	2	0.667
12	2	0.711
13	2	0.698
14	2	0.711
15	2	0.694
16	2	0.676
17	2	0.638
18	2	0.63
19	2	0.638
20	2	0.63
21	2	0.649
22	2	0.615
23	2	0.605
24	2	0.577
25	2	0.568

Table 11. Syntax of `step5.sql` and selected records of output.

```
select   step3.dayno,
         step3.minofflow
from     step3,
         step4
where    step3.segment=step4.segment and
         step4.countofadjsegment=2 and
         0.9*step3.minofflow<step4.minofminofflow
order by step3.dayno ;
```

dayno	*minofflow*
7	0.364
17	0.628
21	0.645
37	0.756
42	0.706
46	0.667
59	0.731
62	0.711
69	0.698
74	0.734
80	0.694
85	0.676
86	0.638
94	0.63
100	0.669
101	0.653
110	0.649
114	0.615
119	0.605
122	0.577
126	0.568
143	0.418
153	0.439
159	0.404
165	0.367

Table 12. Syntax of `step6.sql` and selected records of output.

```
select   step1.dayno,
         step1.flow
from     step1
order by step1.dayno ;

drop table step1 ;
drop table step2 ;
drop table step3 ;
drop table step4 ;
```

dayno	*flow*
1	0.422
2	0.426
3	0.421
4	0.398
5	0.389
6	0.395
7	0.364
8	0.379
9	0.593
10	0.775
11	1.02
12	0.74
13	0.63
14	0.988
15	0.74
16	0.708
17	0.628
18	0.657
19	0.633
20	0.628
21	0.645
22	0.784
23	1.55
24	1.18
25	0.937

Table 13. Syntax of `basebydayno.octave` and selected records of output.

```
load basebydayno.temp ;
t1min=min(table1(:,1)) ;
t1max=max(table1(:,1)) ;
n2=rows(table2) ;
j=0 ;
for i=1:n2
  if(table2(i,1) >= t1min && table2(i,1) <= t1max)
    j=j+1 ;
    table3(j,1)=table2(i,1) ;
    table3(j,2)=min(lagrange(table1,table2(i,1),1),table2(i,2)) ;
  endif
end
format free ;
disp(table3) ;
```

dayno	*base*
7	0.364
8	0.379
9	0.4168
10	0.4432
11	0.4696
12	0.496
13	0.5224
14	0.5488
15	0.5752
16	0.6016
17	0.628
18	0.63225
19	0.633
20	0.628
21	0.645
22	0.651938
23	0.658875
24	0.665812
25	0.67275
26	0.679688
27	0.686625
28	0.693562
29	0.7005
30	0.707438
31	0.714375

Table 14. Syntax of lagrange.m.

```
function y0 = lagrange(tab,x0,N);
%LAGRANGE  Lagrange interpolation of arbitrary order. Y = LAGRANGE(TAB,X0,N)
%          returns an N-th order interpolated value from table TAB, looking
%          up X0 in the first column of TAB.

%          NOTE: TAB's 1st column is checked for monotonicity. It is an
%          error to request a value outside the range of the first column
%          of TAB for X0.

%          Michael F. Saucier 10-16-87

if (nargin ~= 3), error('Wrong number of input arguments.'), end

dx = diff(tab(:,1));
sig = sign(dx(1));
if any(sign(dx)-sig),
  error('First column of the table must be monotonic.')
end

i = find(tab(:,1) == x0);
%if i ~= 0, y0 = tab(i,2); return, end
if (~ isempty(i)), y0 = tab(i,2); return, end %DT

[m,n] = size(tab);
jmin = min(max(min(find(tab(:,1) > x0)) - fix((N+1)/2),1),m-N);
tab2 = tab(jmin:jmin+N,:);
jj = 1:N+1;

seq = x0*ones(1,N+1) - tab2(jj,1)';
lnum = prod(seq) ./ seq;

lden = ones(1,N+1);
for i=jj,
  for j=jj,
    if j~=i, lden(i) = lden(i) * (tab2(i,1)-tab2(j,1)); end
  end
end

y0 = sum(lnum' ./ lden' .* tab2(jj,2));
```

Table 15. Structure and content of the `basebydayno` table.

station	*dayno*	*base*	*station*	*dayno*	*base*	*station*	*dayno*	*base*
PANG	7	0.364	PANG	42	0.706	PANG	77	0.714
PANG	8	0.379	PANG	43	0.69625	PANG	78	0.697
PANG	9	0.4168	PANG	44	0.6865	PANG	79	0.700667
PANG	10	0.4432	PANG	45	0.67675	PANG	80	0.694
PANG	11	0.4696	PANG	46	0.667	PANG	81	0.686
PANG	12	0.496	PANG	47	0.671923	PANG	82	0.6868
PANG	13	0.5224	PANG	48	0.676846	PANG	83	0.6832
PANG	14	0.5488	PANG	49	0.681769	PANG	84	0.6796
PANG	15	0.5752	PANG	50	0.686692	PANG	85	0.676
PANG	16	0.6016	PANG	51	0.691615	PANG	86	0.638
PANG	17	0.628	PANG	52	0.696538	PANG	87	0.637
PANG	18	0.63225	PANG	53	0.701462	PANG	88	0.636
PANG	19	0.633	PANG	54	0.706385	PANG	89	0.635
PANG	20	0.628	PANG	55	0.711308	PANG	90	0.634
PANG	21	0.645	PANG	56	0.716231	PANG	91	0.633
PANG	22	0.651938	PANG	57	0.721154	PANG	92	0.632
PANG	23	0.658875	PANG	58	0.726077	PANG	93	0.631
PANG	24	0.665812	PANG	59	0.731	PANG	94	0.63
PANG	25	0.67275	PANG	60	0.724333	PANG	95	0.6365
PANG	26	0.679688	PANG	61	0.717667	PANG	96	0.643
PANG	27	0.686625	PANG	62	0.711	PANG	97	0.6495
PANG	28	0.693562	PANG	63	0.709143	PANG	98	0.656
PANG	29	0.7005	PANG	64	0.707286	PANG	99	0.6625
PANG	30	0.707438	PANG	65	0.705429	PANG	100	0.669
PANG	31	0.714375	PANG	66	0.703571	PANG	101	0.653
PANG	32	0.721313	PANG	67	0.701714	PANG	102	0.652556
PANG	33	0.72825	PANG	68	0.699857	PANG	103	0.652111
PANG	34	0.735187	PANG	69	0.698	PANG	104	0.651667
PANG	35	0.742125	PANG	70	0.7052	PANG	105	0.651222
PANG	36	0.749062	PANG	71	0.7124	PANG	106	0.650778
PANG	37	0.756	PANG	72	0.7196	PANG	107	0.650333
PANG	38	0.746	PANG	73	0.7268	PANG	108	0.649889
PANG	39	0.736	PANG	74	0.734	PANG	109	0.649444
PANG	40	0.726	PANG	75	0.727333	PANG	110	0.649
PANG	41	0.716	PANG	76	0.720667	PANG	111	0.6405

Table 15. Structure and content of the basebydayno table (continued).

station	*dayno*	*base*	*station*	*dayno*	*base*	*station*	*dayno*	*base*
PANG	112	0.632	PANG	147	0.4264	PANG	182	0.3465
PANG	113	0.6235	PANG	148	0.4285	PANG	183	0.34075
PANG	114	0.615	PANG	149	0.4306	PANG	184	0.335
PANG	115	0.613	PANG	150	0.4327	PANG	185	0.32925
PANG	116	0.611	PANG	151	0.4348	PANG	186	0.3235
PANG	117	0.609	PANG	152	0.4369	PANG	187	0.31775
PANG	118	0.607	PANG	153	0.439	PANG	188	0.312
PANG	119	0.605	PANG	154	0.433167	PANG	189	0.311333
PANG	120	0.595667	PANG	155	0.427333	PANG	190	0.310667
PANG	121	0.586333	PANG	156	0.4215	PANG	191	0.31
PANG	122	0.577	PANG	157	0.415667	PANG	192	0.309333
PANG	123	0.57475	PANG	158	0.409833	PANG	193	0.308667
PANG	124	0.5725	PANG	159	0.404	PANG	194	0.308
PANG	125	0.57025	PANG	160	0.397833	PANG	195	0.3144
PANG	126	0.568	PANG	161	0.391667	PANG	196	0.3208
PANG	127	0.559176	PANG	162	0.3855	PANG	197	0.3272
PANG	128	0.550353	PANG	163	0.379333	PANG	198	0.3336
PANG	129	0.541529	PANG	164	0.373167	PANG	199	0.34
PANG	130	0.532706	PANG	165	0.367	PANG	200	0.34
PANG	131	0.523882	PANG	166	0.37	PANG	201	0.34
PANG	132	0.515059	PANG	167	0.373	PANG	202	0.34
PANG	133	0.506235	PANG	168	0.376	PANG	203	0.34
PANG	134	0.497412	PANG	169	0.379	PANG	204	0.34
PANG	135	0.488588	PANG	170	0.382	PANG	205	0.3338
PANG	136	0.479765	PANG	171	0.3685	PANG	206	0.3276
PANG	137	0.470941	PANG	172	0.355	PANG	207	0.3214
PANG	138	0.462118	PANG	173	0.355375	PANG	208	0.3152
PANG	139	0.453294	PANG	174	0.35575	PANG	209	0.309
PANG	140	0.444471	PANG	175	0.356125	PANG	210	0.3028
PANG	141	0.435647	PANG	176	0.3565	PANG	211	0.2966
PANG	142	0.426824	PANG	177	0.356875	PANG	212	0.2904
PANG	143	0.418	PANG	178	0.35725	PANG	213	0.2842
PANG	144	0.4201	PANG	179	0.357625	PANG	214	0.278
PANG	145	0.4222	PANG	180	0.358	PANG	215	0.2795
PANG	146	0.4243	PANG	181	0.35225	PANG	216	0.281

Table 15. Structure and content of the `basebydayno` table (continued).

station	*dayno*	*base*
PANG	217	0.282889
PANG	218	0.284778
PANG	219	0.286667
PANG	220	0.288556
PANG	221	0.290444
PANG	222	0.292333
PANG	223	0.294222
PANG	224	0.296111
PANG	225	0.298
PANG	226	0.29625
PANG	227	0.2945
PANG	228	0.29275
PANG	229	0.291

PRINTED IN CANADA
IMPRIMÉ AU CANADA
ON RECYCLED PAPER
SUR DU PAPIER RECYCLÉ

National Water Research Institute
Environment Canada
Canada Centre for Inland Waters
P.O. Box 5050
867 Lakeshore Road
Burlington, Ontario
L7R 4A6 Canada

National Hydrology Research Centre
11 Innovation Boulevard
Saskatoon, Saskatchewan
S7N 3H5 Canada

NATIONAL WATER RESEARCH INSTITUTE
INSTITUT NATIONAL DE RECHERCHE SUR LES EAUX

Institut national de recherche sur les eaux
Environnement Canada
Centre canadien des eaux intérieures
Case postale 5050
867, chemin Lakeshore
Burlington, Ontario
L7R 4A6 Canada

Centre national de recherche en hydrologie
11, boul. Innovation
Saskatoon, Saskatchewan
S7N 3H5 Canada

Environment Canada Environnement Canada

Canada