DNMI

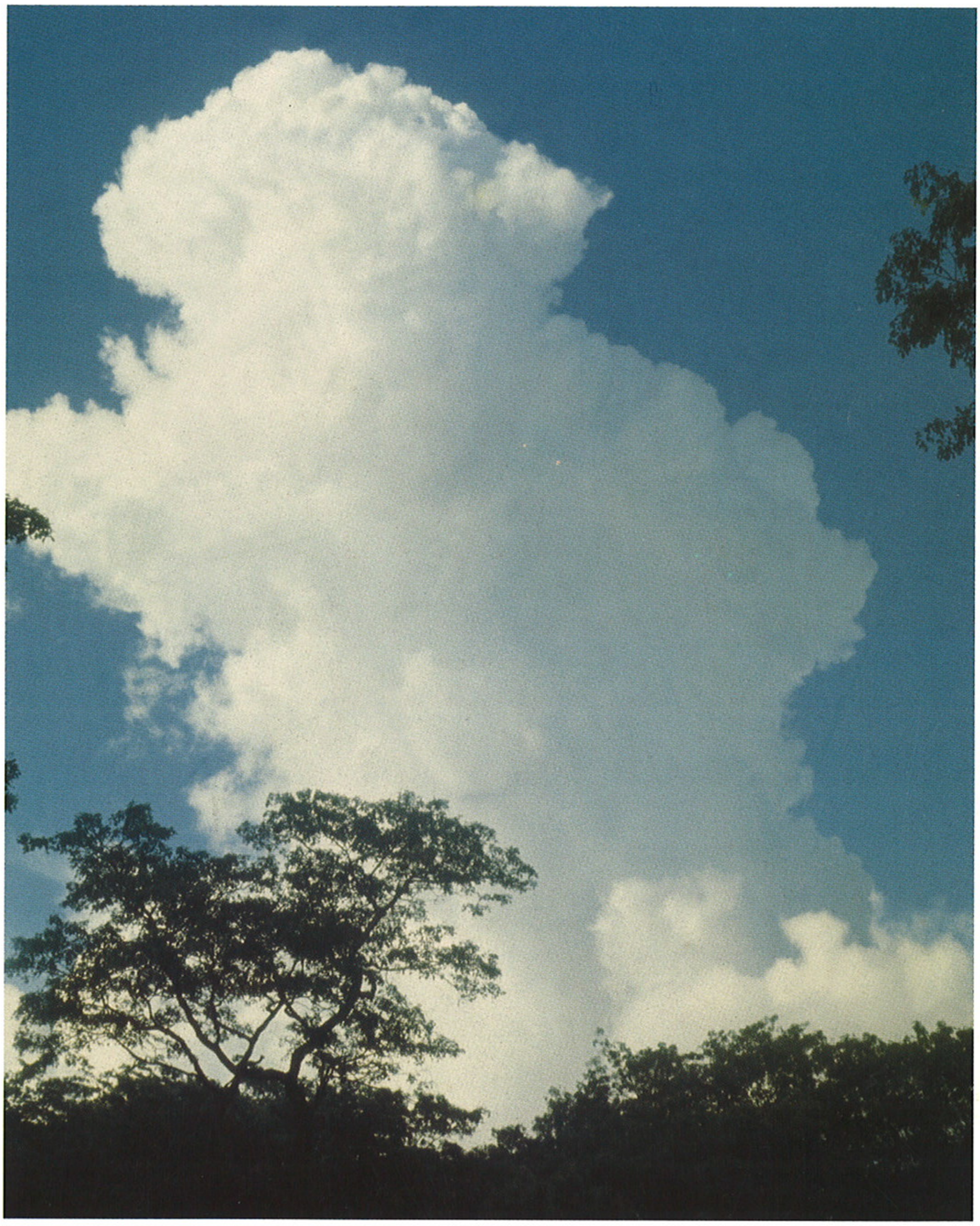
Det norske meteorologiske institutt

KLIBAS research notes volume 3

Report no. 08/99

Petter Øgland (ed.)

KLIMA



DNMI - REPORT

NORWEGIAN METEOROLOGICAL INSTITUTE
BOX 43 BLINDERN N-0313 OSLO

PHONE: +47 22 96 30 00

ISSN 0805-9918

REPORT NO.

08/99 KLIMA

DATE

Mar 1 1999

TITLE

KLIBAS RESEARCH NOTES VOLUME 3

AUTHOR

Petter Øgland (ed.)

PROJECT CONTRACTOR

DNMI - Climatology Division / NORDKLIM

SUMMARY

In this volume 16 research notes dating from February 1999 are presented. All notes are related to on-going research and development of the KLIBAS climatological database system.

The problems analysed split into seven categories.

1. Updates and preparations in order to migrate the current KLIBAS database system from the Oracle7 database on SGI-gale onto the Oracle8 database on SGI-thunder.
2. Quality control in the ALV data processing routine.
3. The XVIND automatic weather stations (VIND_REG).
4. The TELE data processing routine.
5. The PIO data processing routine.
6. The automatic weather stations (AWS).
7. The ALN data processing routine.

KEYWORDS
1. Climatological databases
2. Meteorological data collection
3. Meteorological quality control
4. ALV, ALN, TELE, PIO, XVIND

SIGNATURE

Petter Øgland	Bjørn Aune
Research Scientist	Head of Climatology Division

Table of Contents

Foreword

This third volume of KLIBAS research notes contains a documentation of problems studied in February 1999. While most of the problems addressed in this collection of notes have later been solved, the notes only contain an analysis of the problem, sometimes with suggestions on how to solve it or alter the situation in order to eliminate the problem in its present form.

The notes are presented in order of writing. Seven areas of climatological database research have been targeted for problem analysis.

1. *Updates and preparations in order to migrate the current KLIBAS database system from the Oracle7 database on SGI-gale on to the Oracle8 database on SGI-Thunder.* According to plans in [1], KLIBAS is long due to be running in an Oracle8 environment on the new SGI computer Thunder. In this collection of notes, two papers are discussing problems with getting present software to run in the new environment. The first note addresses the problems with updates of user enviroment and first attempts of getting the Pro*C compiter working. The second note discusses the Pro*C compiler in greater detail.

2. *Quality control in the ALV data processing routine.* The first note discusses adjusting logical tests in the CONTSYN1 quality control program. In the second note problems having to do with the air pressure quality controls in the CONTSYN2 program are discussed. This discussion resulted in an improved version 2.0 of CONTSYN2, documented in [2]. The discussions also contributed to the related problem of how to develop an automatic analysis of the CONTSYN2 output for improving the ALV data processing routine, see [3].

3. *The XVIND automatic weather stations (VIND_REG).* A couple of aspects of the XVIND routine were analysed during this month. The first note is on solving a problem having to do with database restrictions in the XVIND table. The second note is a special analysis on what to do when the VIND_REG program breaks down with a message about being unable to open datafiles.

4. *The TELE data processing routine.* There are four notes concerning the TELE data processing routine this month. The first note deals with preparations needed before adjusting daily interpolated meteorological values with monthly normals. The conclusion of this paper ended with the definition of the next, namely using HIRLAM forecast values as estimates in an automatic interpolation routine for the TELE data processing. Such a system was then designed and implemented [4].

   Another related problem, discussed some what later, was the improvement of the INTERPOL_P0 air pressure interpolation routine. Suggested improvements were then implemented [5].

   The final note on the TELE routine, however, was a specific analysis of problems with the automatic weather station at Finsevatn causing it to top the error statistics for the CHECK_H_STAT quality control program. The analysis resulted in an update of the CHECK_H_STAT program [6].

5. *The PIO data processing routine.* Two problems were discussed this month. The first note describes a problem with reading cloud cover N for PIO observations into the KLIBAS database system. The other note is about a problem with reading snow depth SS and ground description E' correctly into the PIO data table. Unrelated to these problems, a status report on the PIO data collection system was also published this month [7].

6. *The automatic weather stations (AWS).* A note was written on modifying a couple of qualiy checks in the ADK quality control system.

7. *The ALN data processing routine.* In order to prepare for further automation of the ALN data processing routine, a note on designing a system for automatic monitoring of the present routine was

made. Updates of monitoring software were sufficiently made [8].

The notes in this volume document a number of problems in the KLIBAS database system during February 1999. The reason why these particular problems were being addressed this month had to do with the automatically updated priority list presented in the status report for January 1999 [9].

Petter Øgland
Blindern, March 1st 1999

References:

[1] Å.M. Vidal and P.O. Kjensli, *Videreutvikling av DNMIs databaser m/grensesnitt.* DNMI, Oslo, March 1998.

[2] P. Øgland, *CONTSYN2 v.2.0: Some additional checks for the ALV routine.* KLIBAS-report no. 07/99, DNMI, Oslo, February 1999.

[3] P. Øgland, *Automated air pressure quality control by CHECK_CONT2.* KLIBAS-report no. 10/99, DNMI, Oslo, February 1999.

[4] P. Øgland, *Interpolating in TELE with Hirlam10 forecast data.* KLIBAS-report no. 08/99, DNMI, Oslo, February 1999.

[5] P. Øgland, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.2.* KLIBAS-report no. 11/99, DNMI, Oslo, February 1999.

[6] P. Øgland, *Monitoring KA_H_STAT with CHECK_H_STAT v.1.1.* KLIBAS-report no. 14/99, DNMI, Oslo, February 1999.

[7] M. Moe, *PIO STASJONER. Status februar 1999.* KLIMA-report no. 07/99, DNMI, Oslo, February 1999.

[8] P. Øgland, *Collecting and monitoring PRECIP system statistics. V.1.1.* KLIBAS-report no. 15/99, DNMI, Oslo, February 1999.

[9] P. Øgland, *Interpolation and Quality Control status for January 1999.* KLIBAS-note no. 03/99, DNMI, Oslo, February 1999.

Adjusting logical tests in the CONTSYN1 data check of the ALV control routine at the Climatology Division at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 2nd, 1999)

ABSTRACT

After the release of the 3.0 version of CONTSYN1 a problem occured with too many errors with the height of the lowest cloud level (H). Soon after a similar problem with to many errors with levels of heavy sea (S). In this note the ideas behind the tests are discussed, suggesting how they should be applied in order to signal warnings in a correct manner by modifying the present code.

Evolution of the CONTSYN1 program

The CONTSYN1 program is a missing value test which is a part of the ALV quality control routine first described in [1] as it was working on the ND-computers and later implemented on the SGI computers in 1995 as described in [2].

The purpose of the CONTSYN1 program, according to [2], is to pin-point situations where either of the elements TT, UU, DD, FF, FB, FX, N, P0, A, PP, S, FG, WW, W1, W2, TN, TX, TG, TW. Cases of EM = 0, H is missing and NH <> 9 and the number of days with RR > 0 are also noticed.

The program works as a supplement to CONTSYN2 that also lists missing values. CONTSYN1 is only run by the person in charge of the ALV routine, however, and problems spotted by the program usually result in one of the following two actions:

A1. An observation has been punched in the wrong column or for a wrong date. The value is moved to the right place.

A2. The observation is missing in the database and also missing on paper. The interpolation expert is then notified in order to fill in an appropriate value.

In 1996 the CONTSYN programs were improved as a version 1.1 of the system became available [3]. The update of the total CONTSYN system included a revision 1.1 of CONTSYN1.

A further update v.2.0 was made of CONTSYN1, documented in [4], but this version contained errors that were difficult to correct, and the revised version was not used after a short initial period.

In July 1998 work on how to redesign and improve the KLIMA quality control routine commenced, and a program KLIMA_KONTR running the CONTSYN programs in simulation mode was established. The system is described in [5].

The present version of CONTSYN1, however, was released in December 1998. This revision 3.0 of CONTSYN1 was totally written from scratch as previous versions of the program could not handle observations from Semi-Automatic Weather Stations (SAWS) [6].

Problems with height to lowest cloud level H

On January 13th 1998 the test for lowest cloud level H was temporarily removed in order to be reinserted after giving an analysis on how and why the test should be applied.

The algorithm applied in the version 3.0 of CONTSYN1 was an interpretation of the theoretical algorithms described in [3]. This particular algoritm says that for each hour of the day 0, 6, 12 and 18 UTC, if H is missing and NH is different from 9 then a warning should be written.

When checking this with the implementation in version 3.0, it seems, however, that the condition about NH being different from 9 had not been included. The test should hence be brought to working condition by adding this condition.

Problems with levels of heavy sea S

A problem reported on January 26th 1998 had to do with the parameter S being reported all to often. By closer examination, it became clear that the reason for the numerous accounts of S had do to with non of the semi-automatic weather stations (SAWS) reading S at 0 UTC.

As the version 3.0 of CONTSYN1 contains a special treatment of the SAWS, this particular problem should be solved by ignoring the 0 UTC observation in case of S for all SAWS.

Further development of CONTSYN1

Apart from correcting whatever may show to be wrong in the present version of the CONTSYN1 program, further development of the system should be related to the development of the ALV routine.

A natural next move for CONTSYN1 would perhaps be to run the program on a daily basis and add functions to it in order to count the number of problems being recorded. This would then help focusing on the most problematic stations of the routine, perhaps giving way for automatic updating of the ALV table if the types of problems found by CONTSYN1 can be solved algoritmically.

References

[1] P. Øgland, *Kvalitetskontroll av værstasjonsdata i Klimaavdelingen,* KLIBAS-report no. 23/94, DNMI, Oslo, 1994.

[2] P. Øgland, *Programvare for kvalitetskontroll av klimadata,* KLIBAS-report no. 29/95, DNMI, Oslo, 1995.

[3] P. Øgland, *Rutine for kvalitetskontroll av klimadata. Versjon 1.1,* KLIBAS-report no. 10/96, DNMI, Oslo, 1996.

[4] P. Øgland, *Upgrading of the Contsyn System for Verification of Linke Data. Contsyn v2.0,* KLIBAS-report no. 12/96, DNMI, Oslo, 1996.

[5] P. Øgland, *KLIMA_KONTR: Simulation and control of a quality control system for weather data,* KLIBAS-report no. 24/98, DNMI, Oslo, 1998.

[6] P. Øgland, *Test for completeness in data series by CONTSYN1 v.3.0.* KLIBAS-report no 69/98, DNMI, Oslo, 1999.

Preparing hardware and software update for the KLIBAS climatological database system at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY

(subm. February 3rd, 1999)

ABSTRACT

In order to migrate the KLIBAS climatological database system from the SGI computer galeha to thunderha and update of the Oracle database from version 7 to version 8, tests have to be applied in order to reduce the risico for failure when the actual migration will take place.

Evolution of the KLIBAS system

In the introduction to the KLIBAS research notes volume one [1], there is a short summary of the development of the KLIBAS climatological database system.

In 1989 it was decided that a new climatological database system should be planned and implemented in order to replace the previous solution. A first version of this, running on a SGI computer called Typhoon and an Oracle6 database, was operative for processing of daily precipitation measurements in 1993 [2].

During the second half of 1994, the Oracle database was updated from version 6 to 7 [3].

The next major update of the database system then occured in 1995, when the Oracle database was updated from version 7.0 to version 7.1 [4].

A database collapse mid April 1998 resulted in updating Oracle to version 7.3 and moving the complete KLIBAS to a new SGI computer galeha. The performance of the database programs improved tremendously as a consequence of this, and earlier problems with system breakdown because of lack of disk space was also solved [5].

A plan for updating Oracle to version 8 and updating the hardware further was also published April 1998 [6]. One important step in reaching this goal is to test the C-compiler and make sure that continued programming will function in the new enviroments, see point seven on the report from meeting the appendix to the KLIBAS statistics from November 1998 [7].

Login THUNDERHA

With the aid of the Computer Division, it is now possible to login thunderha for some users, although updates need to be done in the .bashrc file for each new user in order to run not the KLIMADB Oracle database on galeha but the KLIBAS database on thunderha.

The easiest way of updating all relevant information for running KLIBAS correctly is probably by copying the .bashrc file from somebody who is already working on thunderha, provided that this user has all the paths that are needed for running Oracle.

Some special updates will then probably have to be done, such as making sure that the NLS_LANG variable is set to American_America.WE8ISO8859P1 for some users while others need it set to Norwegian_Norway.WE8ISO8859P1.

Testing the Pro*C compiler

The only thing that is needed to make sure that the new KLIBAS environment is ready for operative use is to test whether it is possible to run and alter the KLIBAS programs.

At the moment not all KLIBAS datatables have been inserted in the new Oracle8 database, datatables like TELE and XVIND are for instance not yet available. One table that is available, though, is the ALV datatable.

Unfortunately, there is at present no C-program in KLIBAS that makes use of ALV alone, so in order to check whether the Pro*C compiler seems to work, it is necessary to produce a new program whose only Oracle interface consists of a query about the ALV table.

A natural program to adapt for this purpose would perhaps be the KLIMA program [8].

Results from testing

When trying to run the KLIMA program directly, it immediately breaks down with the warning "Error while trying to retrieve text for error ORA-12545" which means that the warning messages have not been properly established for Oracle8 in this new version.

More importantly, however, is how the program reacts if we try to compile. The message we receive then is "/usr/people/oracle/klimadb/7.3.3/precomp/env_precomp.mk for including".

Conclusions

The Oracle8 database on SGI Thunder is presently not ready for migrating KLIBAS. Both the Oracle message handler and the Pro*C compiler need to be installed correctly and further testing need to be carried out.

Relevant information must be sent to the responsible persons in order to help clear things up.

References

[1] P. Øgland, *KLIBAS research notes volume 1.* KLIMA-report no. 01/99, DNMI, Oslo, 1999.

[2] M. Moe et al, *Databaseprosjektet i Klimaavdelingen. Status pr. 31.12.1993.* KLIBAS-report no. 03/94, DNMI, Oslo, 1994.

[3] P. Øgland, *Omlegging av databaserutiner ved overgang fra Oracle6 til Oracle7.* KLIBAS-report no. 28/94, DNMI, Oslo, 1994.

[4] P. Øgland et al, *Databasegruppen 1995.* KLIBAS-report no. 01/96, DNMI, Oslo, 1996.

[5] P. Øgland, *Maintenance and Performance Statistics April 1998.* KLIBAS-note no. 10/98, DNMI, Oslo, 1998.

[6] Å.M. Vidal and P.O. Kjensli, *Videreutvikling av DNMIs databaser m/grensesnitt.* Klimaavdelingen, DNMI, Oslo, 01.04.1998.

[7] P. Øgland, *KLIBAS Process Improvement November 1998.* KLIBAS-note no. 26/98, DNMI, Oslo, 1998.

[8] P. Øgland, *Collecting KLIMA system statistics.* KLIBAS-report no. 34/98, DNMI, Oslo, 1998.

Solving a problem having to do with database restrictions in the XVIND table for the KLIBAS climatological database system at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 3rd, 1999)

ABSTRACT

The VIND_REG data transport program is responsible for reading automatic wind registrations from data files and inserting these into the XVIND Oracle data table in the KLIBAS database system. This paper is concerned with a new manner in which the program is now breaking down, the problem now being related to check constraints in the XVIND datatable. The problem is discussed, and a solution is suggested.

The VIND_REG program

Specifications for the VIND_REG program were written by Lars Andresen in September 1997, resulting in a first version of VIND_REG being released the 12th of July 1998 [1]. The purpose of the program was to read files, interpret and modify observations when necessary and store the results in suitable datatables in the KLIBAS database.

By August 18th 1998 a routine was being established at the Climatology Division for handling the observations. A quality control program VINDDEK was then released according to specifications by Lars Andresen and P.O. Kjensli [2].

By the end of August the same year, a revised version of VIND_REG appeared [3]. Minor faults in the algorithm had been corrected, the specification had been slightly updated.

Still there were problems with the program, and on the 20th of October 1998 a version 1.2 of VIND_REG was released [4]. The program was now running systemaically every hour of the day by the crontab schedule on the UNIX machine. A new table XVIND was designed for storing the observations.

The first log of the program is in April 1998. It was then run 17 times, failing 0 of these. The curve below shows the relative number of runs of the program that has been failing according to the log.

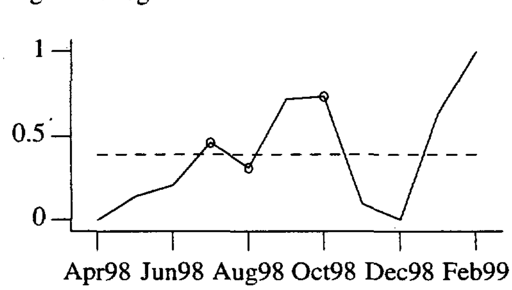


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.39. The time when the revisions of the program appeared are marked with a small circle.

Breakdown because of constraint violation

On Saturday 30 1999 the VIND_REG program collapsed in an attempt to reload data files older than 1998 as these had not yet been stored in the XVIND structure.

The program failed in the procedure "readVINDintoXVIND" with the message "ORA-02290: check constraint (V.VINDCH_RHNR) violated" produced from the Oracle database message handler. The function "readVINDintoXVIND" was called from the main loop of the program.

The specifications of the constraints for the XVIND table are given as an appendix to the version 1.2 description of the system [4], which states that the column RHNR is defined to be an integer value between 1 and 5.

What the program tried to do, however, when reading the file 'enbn0498.txt', containing observations from 76330 Brønnøysund Lufthavn, was to insert a character value in this position.

The reason for failure seem to be that the program misinterprets the final word on each line of the inputfile, reading 'ENBNH' to mean that the station 'ENBN' is of type 'NH', not 'H' as it should.

Before values are inserted into the XVIND table, all values on file are inserted into a temporary table called VIND, where some data processing is done before final storage. The content of the column TYPE in VIND, which is later used for updating the RHTYPE and RHNR columns in XVIND, is normally decided by the function "readIdentifierFromFile" that checks the first line on file. The reason for failure may have to do with how this line is checked.

Presently the complete line is read, and the the final characters are tested against a pattern of legal TYPEs.

Suggested solution

In order to improve on the situation, it would probably be sensible to read this line word by word, and then check the final word. This would at least take care of normal situations where the last word of the line contains the station identification.

In some cases, however, like on the file 'enfg0798.txt', this final word is the default 'WAT15H'. By following the method above, the TYPE column in VIND would contain the word '5H', and the update of XVIND would once again collapse.

In order to prevent this, a check have to be applied so that whenever one of the default type are not recognised, the type 'H' is given as default.

The way VIND_REG have been programmed up to now, it only checks whether the TYPE contains one of the words 'H', 'R1', 'R2', 'F1' or 'F2'. As the word '5H' contains 'H' this is not a good test, and should be altered into a strict comparison.

Special care has also to be taken when the final word is less than four letters long. It is then obvious that the final word is not a descriptor, and TYPE should be given the value 'H' be default.

References

[1] P. Øgland, *Computer program VIND_REG.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program VINDDEK.* KLIBAS-report no. 39/98, DNMI, Oslo, 1998.

[3] P. Øgland, *Computer program VIND_REG v.1.1.* KLIBAS-report no. 43/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

Problems with air pressure checks in the CONTSYN2 data control of the ALV routine at the Climatology Division at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY

(subm. February 4th, 1999)

ABSTRACT

In order to check air pressure in the CONTSYN2 quality control program, tests vary for some particular stations in a manner which makes it necesary to update the code of the program when certain stations particularily located are introduced to the station network. This note addresses a problem that is particular for the station Fiplingsvatn.

The CONTSYN2 program

The CONTSYN2 is a general quality control which is a part of the ALV quality control routine first described in [1] as it was working on the ND-computers and later implemented on the SGI computers in September 1995 as described in [2].

A version 1.1 of the program appeared in December 1995 [3] with some minor adjustments in April 1996 [4].

A version 2.0 of CONTSYN2, presently only including additional tests to the version 1.0 of the program, and run as an appendix to version 1.0, was established in May 1998.

It was then run 184 times, failing 1 of these (0.5%). The number of users during this first month was 9. The curve below shows the relative number of runs of the program that has been failing according to the log.

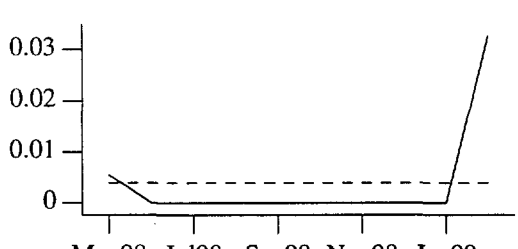


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.38. The only times there have been recorded problems with the program has been during periods of maintenace and update.

The air pressure quality checks

It has been known for a long time that the air pressure quality checks in the v.1.0 CONTSYN2 program are far from optimal. In fact, much effort was put into have these tests updated when the program was reimplemented in 1994 and 1995. At the time, however, other programming tasks were given higher priority.

On the 18th of January 1999, however, the problem with the air pressure algoritms were addressed once again as a new weather station, 77550 Fiplingsvatn, entered the system and produced a significant amount of warnings although operationally working all right.

According to an analysis performed by Lars Andresen, special considerations should be taken when producing warnings for Fiplingsvatn as this station is subject to inversion, and special air pressure reduction formulas have to be applied.

A problem with the v.1.0 CONTSYN2 software, however, is that the source code for this program is no longer available, so all alterations of the code have to be done in terms of making new programs.

Suggestions on how to handle the problem

As a total rework of CONTSYN2 is a major task, the best thing is probably just to add a warning in CONTSYN2 whenever it is run for a station like Fiplingsvatn, explaining that due to inversion, the warnings may be redundant.

The next step would then be to produce the sort of warnings in CONTSYN2 v.2.0 that actually should be reported, and, finally, as all the checks for all elements in CONTSYN2 v.1.0 have been replaced by programs in the v.2 series, the origianl v.1.0 should be put to rest.

References

[1] P. Øgland, *Kvalitetskontroll av værstasjonsdata i Klimaavdelingen,* KLIBAS-report no. 23/94, DNMI, Oslo, 1994.

[1] P. Øgland, *Migrasjon av datakontroll CONTSYN2 fra ND-100/788 til SGI-Typhoon.* KLIBAS-report no. 23/95, DNMI, Oslo, 1995.

[3] P. Øgland, *Programvare for kvalitetskontroll av klimadata,* KLIBAS-report no. 29/95, DNMI, Oslo, 1995.

[4] P. Øgland, *Rutine for kvalitetskontroll av klimadata. Versjon 1.1,* KLIBAS-report no. 10/96, DNMI, Oslo, 1996.

[5] P. Øgland, *KLIMA_KONTR: Simulation and control of a quality control system for weather data,* KLIBAS-report no. 24/98, DNMI, Oslo, 1998.

Preparations needed before adjusting daily interpolated meteorological values with monthly normals at the TELE routine at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo
(subm. February 6th, 1999)

ABSTRACT

In this paper we discuss the present situation with respect to filling in missing observations in the TELE datatable of the TELE routine. We then address the question on how to improve the situation by various different interpolation techniques when the present method is not applicable.

SYNO_KONTR and the TELE data table

The first version of SYNO_KONTR was written as a part of the version 3.0 of SYNO_INN in October 1996 [1]. The purpose of this first program was to compare observations in the datatables SYNOP and SYNOP2 that were both updated by SYNO_INN v.3.0.

In August 1997 the revised verson 2.0 of SYNO_KONTR was released [2]. Now the program was responsible for running several quality control and interpolation programs on a daily basis and generate statistics from the execution of these programs.

A version 2.1 of SYNO_KONTR [3] was then released on January 20th 1999, now including statistics for monitoring different aspects of the TELE table, especially aspects having to do with whether interpolation had been performed successfully or not.

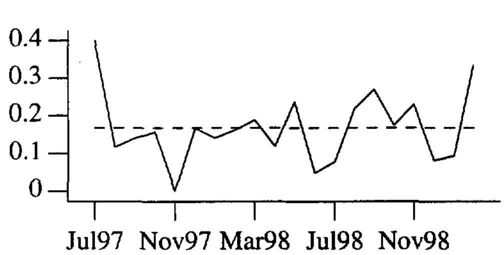


Fig 1. Relative number of failures

The first log of the program is in July 1997. It was then run 15 times, failing 6 of these (40.0%). The curve below shows the relative number of runs of the program that has been failing according to the log.

The average value dashed in the figure no 1. is 0.17. There is no apparent visual indication in the figure that might indicate that the program is stabilising although the two last months show a defect ratio significantly below average.

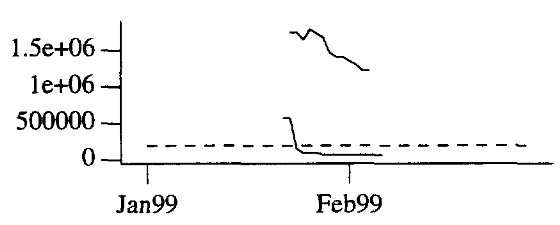


Fig 2. Day by day defects in TELE

In figure 2 the total number of irregularities in TELE, for a special group of weather elements subject to interpolation and automatic manipulation, have been recorded and displayed. The process average is 150360 defects wit a standard deviation of 1730924. As improvements were made late January, confer [3], the 6 sigma process capacity drawn in solid as the upper curve, is rapidily falling as the process is improving as a consequence of program improvements.

The SYNO_KONTR program tries to classify problems whether they have been created by SYNO_INN, INTERPOL2 [4], INTERPOL3 [5], INTERPOL_P0 [6] or manual updates in TELE according to flag information.

At present there are 28933 cases where columns that should contain data for either TT, TN, TX, RR, SS, N or UU but which are empty. Apparently the INTERPOL2 program has not been able to produce estimates for interpolation.

There are 10112 cases where columns that should contain data for either P, P0, PP or A but which are empty. Apparently the INTERPOL_P0 program did not manage to produce estimates for interpolation. Adding this to the problems caused by INTERPOL2 we get a total of 39045.

There are 27851 cases where columns that should be empty according to TELE_PARA contained data with flag=0. This may be explained by either TELE_PARA not updated by the SYNO_INN program [7] or the SYNO_INN program not assigning observations to the correct station perhaps because of errors in the syno files.

There are 1451 cases where columns that should be empty according to TELE_PARA contained data with flag=3, flag=5 or flag=6. Neither of these were expected. The programs VNN/RETTELSER (flag=3), INTERPOL3 (flag=5), ALA2ALV (flag=5) and ALF2TELE (flag=6) should be investigated.

There are 622 cases where columns that should be empty according to TELE_PARA contained data with flag=1 or flag=2, meaning that the observation has been manually updated. Either the observations should not have been updated or TELE_PARA is not up to date.

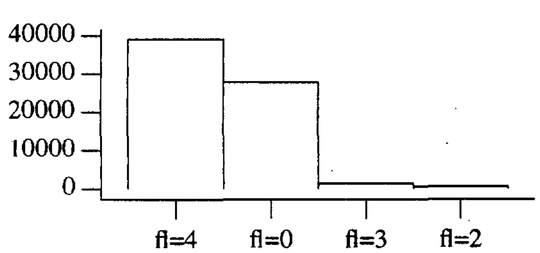


Fig 3. Size of problem

The diagram in figure 3 illustrates that the greatest effect on quality improvement on the content of the TELE data table would probably be to improve the algorithms that should secure that interpolated values are inserted. In other words, if we can improve the interpolation methods, this is the currently best way of improving the general data quality of TELE.

When observations are not interpolated

The program INTERPOL2 is responsible for interpolating missing observations for temperature (TT, TN, TX), relative humidity (UU), precipitation (RR) and cloud cover (N). The program INTERPOL_P0 interpolates air pressure parametres P, P0, PP and A.

Both the programs are designed in order to fill in blanks for all stations defined in TELE according the specifications in TELE_PARA. When no value in TELE occurs, the reason for this may be manyfold.

1. The estimates made by INTERPOL2 or INTERPOL_P0 were so poor, according to internal tests of the the programs, that they were not used by the programs. The reason for this may, for instance, be that there were not sufficient reference data to make estimates, or there were no observations to test against, so it was impossible to decide how good the estimates were.
2. Interpolation may have been performed, but the inserted values were later removed by S-T-F [8] as they were not consistent with quality checks.
3. Some stations may not be interpolated as they are not part of the stations defined when using the MANGELLISTE datacover program [9].

Of these three reasons, the third reason is the only one that is fairly simple to control. The program MANGELLISTE is run by SYNO_KONTR, defining the population of stations in the interaction between the programs. By checking SYNO_KONTR it is verified that the option no. 7, all stations defined in TELE_PARA, is the one used by the program.

This is indeed as expected as, for instance, the non-Norwegian meteorological stations which are interpolated do not occur on any of the other six options available to the program.

The question remains then, how to produce complete data cover when still be restricted to the rules given in the cases one and two above. First of all, however, in order to achieve control, it seems reasonable to reprogram the system so that it is possible to find out if a missing observations is due to case one or case two.

Presently, S-T-F removes both the interpolated value and the corresponding value in the flag column. For instance, if an intolerable value of TT was found, which also has a corresponding FLTT='4', then both TT and FLTT would be assigned the value NULL, that is, values will be removed from both locations.

The most natural way of solving this, perhaps, is to set TT:=NULL and FLTT:='5' as '5' is the value given for automatic updates by the program INTERPOL3.

The real problem, however, is how to fill in values for instances described by case one. How is it by existing methods possible to estimate and interpolate for, say, arctic stations where there are no references stations or no relevant reference data either on the station or on neibouring stations if there are any.

As none of the present methods, or variations over these, seem very relevant for addressing this problem, the best approach may perhaps be to use forecast data as estimates. Preparations for such a program was, however, done in September 1998 [10], based on a discussion with Laila Sisselrud in September 1997.

Before further improvement on the interpolation and quality control systems for the TELE routine may be done, the HIRLAM forecast estimation/interpolation method should be implemented.

References

[1] P. Øgland, *Eksperimentell overføring av data fra syno-filer til Oracle-database. Versjon 3.0.* KLIBAS-report no. 15/96, DNMI, Oslo, 1996.

[2] P. Øgland, *Døgnlig driftskontroll for SYNOP: syno_kontr v.2.0.* KLIBAS-report no. 60/97, DNMI, Oslo, 1997.

[3] P. Øgland, *SYNO_KONTR v.2.1: Daily interpolation and quality control. KLIBAS-report no. 06/99, DNMI, Oslo, 1999.*

[4] P. Øgland, *Automatic interpolation of SYNOP weather observations: Interpol1 and Interpol2.* KLIBAS-report no. 61/97, DNMI, Oslo, 1997.

[5] P. Øgland, *Automatic correction of SYNOP weather observations: Control1, Control2 and Interpol3.* KLIBAS-report no. 62/97, DNMI, Oslo, 1997.

[6] P. Øgland, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.1.* KLIBAS-report no. 52/98, DNMI, Oslo, 1998.

[7] P. Øgland, *Reading data from syno-files into KLIBAS: SYNO_INN v.3.7.* KLIBAS-report no. 05/99, DNMI, Oslo, 1999.

[8] P. Øgland, *Quality control computer program S-T-F v.2.1: Automatic removal of "bad" INTERPOL2 values.* KLIBAS-report no. 17/98, DNMI, Oslo, 1998.

[9] P. Øgland, *Updating data in TELE by VNN v.2.1.* KLIBAS-report no. 62/99, DNMI, Oslo, 1999.

[10] P. Øgland, *KLIBAS statistics September 1998.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

Using HIRLAM10 forecast values as estimates in an automatic interpolation routine for the TELE routine in KLIBAS at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 9th, 1999)

ABSTRACT

In this paper we discuss how to apply forecast data from the HIRLAM10 model for interpolation of missing observations in the TELE data table. The discussion focuses on how the address the HIRLAM10 data, how the estimates should be made and how to implement and evaluate the interpolation program.

Using forecast data for interpolation

During the presentation of current state of the art in precipitation quality control at the Nordic Meteorological conferance in 1994 [1], one of the questions from the panel was whether it might be worth considering using forecast data for quality control purposes.

Even though the idea had been discussed informally also prior to the meeting, it was not until a discussion with Laila Sisselrud at DNMI/FoU in September 1997 that sketches for a program were made, giving rise to a very preliminary beta version of the program in September 1998 [2].

The expected quality of a forecast-interpolation method should be given by the validation results from present research at DNMI. For precipitation at 06 UTC, for the period of September 1-31 1998, the RMSE was 2.5 for Oslo - Blindern on the Hirlam10 model and 3.1 on the Hirlam1 model [3].

Comparing these with the estimates made by the INTERPOL2 program [4], the RMSE for September here is about 0.6 mm [2], although the month is highly incomplete. The statistics for December 1998 [4], represented by curve only, gives an average RMSE of about 1.0 mm.

Anyway, either of the Hirlam models should be useful for generating interpolation values of roughly similar quality to the interpolation values already being used.

How to get HIRLAM results

In order to validate observations, the prognosis data are stored in an Oracle database MIOPDB on the same computer GALEHA that contains the KLIBAS climatological database.

During a discussion with Arnstein Foss at DNMI/FoU, the easiest way of getting to these data seemed to be by login as user guest on GALEHA, the access Oracle as user VERIF, collecting data from tables that are typically described as verifop.hirlam1_1492, the code 1492 meaning synop no. 492, Oslo - Blindern, fo zone no. 1, Norway. Observations seem to have been stored since May 1995.

The table hirlam1_1492 contains the columns AAR, MND, DAG, TIM, PROG, DD, FF, TT, TD, RR, FG, CL, CM, CH, P, N, TT_K where all columns except PROG and TT_K are the ones described for the KLIBAS database [5]. The column PROG contains the number of hours since the prognoses was being made, for Blindern there are 17 values, namely 0, 3, 6, ..., 48, and TT_K is the temperature from the model modified by Kalman filter techniques.

For interpolation purposes, the elements TT, TD, RR, P and N are of special interest. For interpolation TT in TELE, either TT or, perhaps even better TT_K should be used, relative humidity UU in TELE should be estimated by TT and TD, precipitation may be updated directly as with could cover N and air pressure at sea level P. Air pressue at station level P0 should then be computed by use of formulae already being used by INTERPOL_P0 [6].

The only elements presently interpolated by INTERPOL2 that is not being handled by HIRLAM10 is TN and TX. These have to be estimated by courses means then. Similarily, air pressure parametres PP and A, interpolated by INTERPOL_P0, can not be inserted directly into TELE by MIOPDB, and some kind of course estimation has to be performed.

For a first version of HIRLAM10 forcast interpolation, however, it should be sufficient to insert TT, RR, N and P as neither of these demand special care except reading for MIOPDB and inserting in TELE/KLIMADB.

Creating a database link

As values are stored in two different databases, we have to establish a link between these. Oracle7 supports a link feature which makes it possible to read from one database into another [7].

The purpose of the HIRLAM program would then be to nest all relevant stations in the TELE routine, the stations that are defined by TELE_PARA and are missing observations for relevant parametres, and the systematically updated TELE for each missing value by reading observation from a corresponding MIOPDB data table.

In fact, a database link is about to be created in order to make the two databases communicate, and constructing SQL sentences that describe the update should be possible.

When the database link is up and going, it should be easy to make a prototype version of the new interpolation program, starting, perhaps with a single SQL command for updating RR in all relevant tables in TELE, then expanding to TT, P and N, and finally adding algorithms for producing TN, TX, UU and P0.

References

[1] P. Øgland, Some Recent Developments Regarding Quality Control of Precipitation Data in Norway. *Preprints av foredrag DET 19. NORDISKE METEOROLOGMØTE.* Kristiansand, June 6.-10. 1994.

[2] P. Øgland, *KLIBAS statistics September 1998.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[3] H. Haakenstand and M.H. Jensen, *Validering av operasjonelle modeller for september, oktober og november 1998.* Research Note no. 21, ISSN 0332-9879, DNMI, Oslo, December 1998.

[4] P. Øgland, *Automatic interpolation of SYNOP weather observations: Interpol1 and Interpol2.* KLIBAS-report no. 61/97, DNMI, Oslo, 1997.

[5] P. Øgland et al., *Utarbeiding og testing av ulike datastruktur på Typhoon. Delprosjekt 2.* KLIMA-report no. 42/92, DNMI, Oslo, 1992.

[6] P. Øgland, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.1.* KLIMA-report no. 52/98, DNMI, Oslo, 1998.

[7] B. Linden et al., *ORACLE7 Server. SQL Language Reference Manual.* Part no. 778-70-1292, Oracle Corporation, December 1992.

Some contributions to an automated analysis of the CONTSYN2 output for improving the ALV routine at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 12th, 1999)

ABSTRACT

The program CHECK_CONT2 is designed for analysing and producing evaluation statistics for stations based on output from the CONTSYN2 quality control program that is part of the ALV quality assurance routine. In this paper we investigate strategies for analysing output relating to air pressure as this is one of the weather elements the present version of CONTSYN2 has some difficulty dealing with.

The CHECK_CONT2 program

In May 1993 a report on quality control methods for the KLIBAS database system was published [1], mostly devoted to precipitation stations, but also with treatment of other types of weather stations, stressing the need of a modular system.

In October 1994 special concern was taken in order to investigate the present state of the ALV quality control routine [2]. The report gave a short historical overview of the development of the quality control routine, a description of how the present system was working, analysing tests and output for all programs, and some suggestions on further development.

By Septebmer 1995, a new ALV quality control was under production for the KLIBAS database environment, starting by migrating the core quality control program CONTSYN2, and making this work under the new environment [3], soon followed by a complete migration of the rest of the system [4].

By April 1996 a version 1.1 of the ALV quality control system was running [5], now debugged to be working with the same stability as it had been doing on the ND-computer system prior to KLIBAS, although some minor attempts at improving the system even further was also made this year [6].

In July 1998, however, a new approach to improvement of the ALV routine was made by running a simulation program KLIMA_KONTR [7] that started subprograms analysing the different results from the ALV quality control routine and made a total evaluation of the system.

Two vital components in the KLIMA_KONTR system were the CHECK_RELFUKT [8] and CHECK_KONTHUM programs, both producing automatic analysis of the output from the quality control programs that were specially designed for locating problems related to realtive humidity, the KLIMA_KONTR system further improved during August 1998 [10].

In December 1998 new work commenced on the CONTSYN1 quality control program as the ALV routine was expanded in order to also handle semi-automatic weather stations (SAWS) [11], and in February 1999 a version 2.0 of CONTSYN2 was made [12] that would produce an appendix to the results produced by the earlier CONTSYN2.

The program CHECK_CONT2 uses output both from CONTSYN2 v.1.1 and v.2.0 in order to investigate the status of each particular station. The first log for this program was produced in

August 1998. It was then run 53 times, failing 2 of these (3.8%).The curve below shows the relative number of runs of the program that has been failing according to the log.

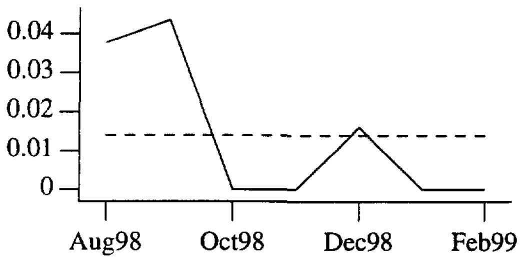


Fig 1. Relative number of failures

The average value dashed in the figure is 0.014. There has been very little trouble with the program in its present version. The program is designed to be run twice a day, executed from KLIMA_KONTR.

The air pressure problem

One straining problem with the CONTSYN2 output is the air pressure control formulas where special cases are hardcoded into the system. As the station network is updated and altered this may result in new or old stations falling into wrong categories, sometimes making the CONTSYN2 output indicate that there may be problems with stations that actually are working as specified.

Similarily, air pressure faults may not be detected if formulas are not set up in a proper manner. Lars Andresen has done a recent analysis of air pressure at some ALV weather stations [11] where he concludes that there may be problems with inversion on certain stations. Fiplingsvatn should use inversion formula, so should Kautokeino, but Byglandsfjord should not. All three stations generate misguided output on CONTSYN2.

Automatic analysis of the CONTSYN2 output

The purpose of the CHECK_CONT2 program is to make an automatic analysis of the CONTSYN2 output in order to help the ALV routine in how to handle quality control problems, and also, as the system develops, make automatic updates in the ALV data table for simple routine errors that are detected by the program and which may be algoritmically detected and solved.

Presently, what the program does, is to loop all stations and count the number of defects or warnings for each particular one, producing statistics and curves that are then printed in the monthly interpolation and quality control status reports [12].

A natural approach, then, on how to solve the problem with air pressure warnings in CONTSYN2 would be:

1) Make the CHECK_CONT2 program do an automatic analysis on the air pressure values, trying to pinpoint what may be the cause of error for each particular point.

2) Add improved quality control checks in the new second version of the program that will eventually replace CONTSYN2 v.1.1 completely.

The first of these two points is imperative in order to improve the system. Naturally, a complete analysis of air pressure is a difficult task to perform, and making CHECK_CONT2 aspire to this is ambitious. A simple version, however, should be relatively easy to implement, checking the results produced by CONSYN2 against internal checks in CHECK_CONT2 based on the revised formulae.

Later, an update of CONTSYN2 v.2.0 would then be a simple enough, just duplicating the tests implemented for verification in CHECK_CONT2 and make the results presentable in an easy manner.

If CONTSYN2 and CHECK_CONT2 do exactly the same analysis, then the test module in CHECK_CONT2 may be removed, in order to have the program just performing a problem identification analysis, leaving the task of checking for errors to CONTSYN2.

References

[1] P.O. Kjensli, S.L. Lystad and P. Øgland, *Kontroll av data - spesifikasjonsrapport.* KLIBAS-report no. 01/93, DNMI, Oslo, 1993.

[2] L. Håland and P. Øgland, *Kvalitetskontroll av værstasjonsdata i Klimaavdelingen.* KLIBAS-report no. 23/94, DNMI, Oslo, 1994.

[3] P. Øgland, *Migrasjon av datakontroll CONTSYN2 fra ND-100/788 til SGI-Typhoon.* KLIBAS-report no. 23/95, DNMI, Oslo, 1995.

[4] P. Øgland, *Programvare for kvalitetskontroll av klimadata.* KLIBAS-report no. 29/95, DNMI, Oslo, 1995.

[5] P. Øgland, *Rutine for kvalitetskontroll av klimadata. Versjon 1.1.* KLIBAS-report no. 10/96, DNMI, Oslo, 1996.

[6] P. Øgland, *Upgrading of the Contsyn System for Verification of Linke Data. Contsyn v2.0.* KLIBAS-report no. 12/96, DNMI, Oslo, 1996.

[7] P. Øgland, *KLIMA_KONTR: Simulation and control of a quality control system for weather data.* KLIBAS-report no. 24/98, DNMI, Oslo, 1998.

[8] P. Øgland, *Computer program CHECK_RELFUKT.* KLIBAS-report no. 37/98, DNMI, Oslo, 1998.

[9] P. Øgland, *Analysing Relative Humidity Statistics by CHECK_KONTHUM.* KLIBAS-report no. 38/98, DNMI, Oslo, 1998.

[10] P. Øgland, *Computer program KLIMA_KONTR v.1.1.* KLIBAS-report no. 40/98, DNMI, Oslo, 1998.

[11] L. Andresen, *Program for beregning av lufttrykk i havets nivå.* Excell output plus handwritten note to Petter Øgland. DNMI/Klima, 22.01.1999.

[12] P. Øgland, *Interpolation and Quality Control status for January 1999.* KLIBAS-note no. 03/99, DNMI, Oslo, 1999.

Automatic interpolation of air pressure at DNMI by the KLIBAS program INTERPOL_P0

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 16th, 1999)

ABSTRACT

Three automatic interpolation programs are running presently in the KLIBAS database. One program HIRLAM fills in missing values with estimates generated by the forecast models, one program INTERPOL2 runs interpolation, except for air pressure, by use of weighted averages, and one program is especially designed for air pressure interpolation, INTERPOL_P0, using radial basis functions. In this paper the INTERPOL_P0 air pressure algoritms are discussed with emphasis on how to improve quality.

The INTERPOL_P0 program

The first attempts at automatic interpolation in TELE were done in June 1994 [1], but this was only a test and the code for this particular program was not developed into full maturity.

The first version of the present interpolation system, INTERPOL2, was released in August 1997 [2]. The interpolation technique was evaluated [3] and then improved [4].

In May 1998 the interpolation program was made to communicate with the quality control program S-T-F by having S-T-F remove observations inserted into TELE by INTERPOL2 when they were in conflict with the S-T-F quality control [5].

In order to improve further on the interpolation system, a special module for interpolating air pressure, INTERPOL_P0, was constructed in September 1998 [6]. This program is running in sequence with INTERPOL2 in order to compare the quality of the estimates done by the two programs, although only estimates from INTERPOL_P0 are inserted into TELE. A revised version of INTERPOL_P0 was released by the end of September 1998, using an improved strategy for interpolating air pressure [7].

All interpolation and quality control programs are run systematically twice a day from a program SYNO_KONTR which is also responsible for checking the content of the TELE data table after all updates have been done [8].

More recently, interpolation by use of Hirlam10 forecast data has also been a part of the automatic TELE interpolation scheme [9].

The first log for INTERPOL_P0 is from August 1998. It was then run 96 times, failing 11 of these (11.5%). The curve below shows the relative number of runs of the program that has been failing according to the log.

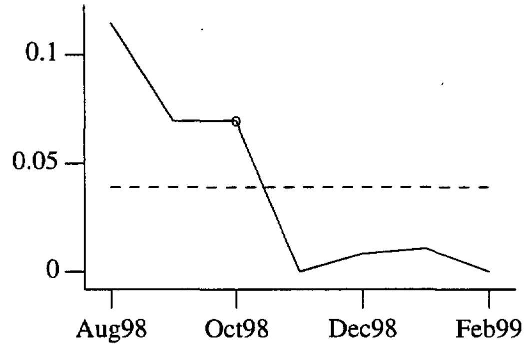


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.04. The program has been fairly stable since November 1998. The revision 1.1 of INTERPOL_P0, on September 29th 1999, is marked by a circle.

Possibilities for improvement

Below is an example of the type of statistics that is produced for the KLIBAS statistics report.

The solid curve represents the root mean square error (RMSE) of the INTERPOL_P) method (I0), while the dashed curve is the RMSE of the INTERPOL2 method (I2). If the INTERPOL_P0 is better than the INTERPOL2 method, the solid curve should show lower values than the dashed curve.

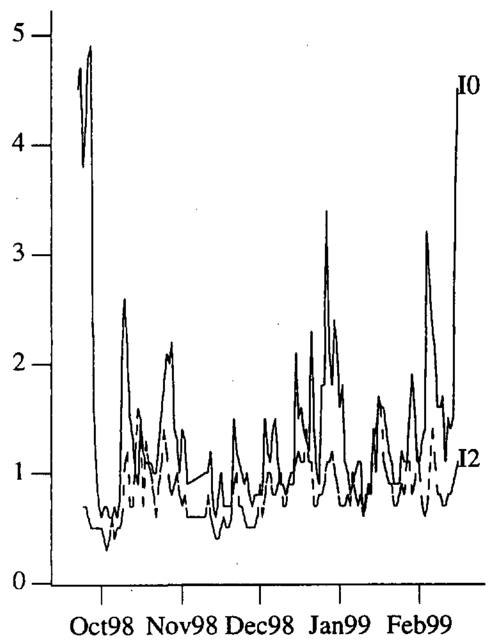


Fig 2. RMSE(p) for INTERPOL_P0 and INTERPOL2

The first drop in the I0 curve at the end of September 1999 is a consequence of improving the program, documented in [7] as the update to version 1.1. The update seem to have had a significant consequence in improving the method, although high RMSE's do still occur.

On the other hand, from figure 2 it is quite apparent that when estimating air pressure at sea level, P, the INTERPOL_P0 method is not any better than the earlier INTERPOL2 method. In fact, the figure shows that the RMSE for INTERPOL_P0 is larger than the RMSE for INTERPOL2 almost all the time, often significantly larger too.

The only reason for using the INTERPOL_P0, then, is that P0 is computed from P in this program, not estimated by a seperate method as it was done in INTERPOL2.

On the other hand, the reason for estimates made by INTERPOL_P0 being no better than the ones produced by INTERPOL2 may have to do with INTERPOL_P0 using radial basis functions for estimation, the method originally used by INTERPOL2, see [3], while the INTERPOL2 model was updated with a specially designed statistical method explained in [4].

The question is, then, whether it is possible to improve the method used in INTERPOL_P0 in order to get results that are even better than the ones produced by INTERPOL2.

Ideas for improving INTERPOL_P0

The most apparent fault with the present method of estimating air pressure at sea level accoring to neighbour stations, is that no consideration for trends in the data set for the test station is taken.

Three ideas should be considered:

1. The most natural step, perhaps, in order to try to improve upon the current sitation would be to us information p(j-1) and p(j+1) in order to estimate p(j), not only q1(j), q2(j), ..., qn(j) for the n reference stations.
2. Another approach would be using air pressure tendency pp(j-1), pp(j) and pp(j+1) to give an estimation of the rate between two consequtive observations of p or p0 and use this rate when estimating.
3. Also, quality control for air pressure should be added to the current S-T-F quality control program, in order to remove the worst estimates and to guide further improvements of the estimation techniques to be consistent withing a quality control regime.

All of the three ideas above should be implemented, perhaps starting with the first suggestion as interpolation values that are often removed from TELE come as spikes in the data set, obviously the current method lacking very much in terms of the first idea described above.

If the methods turns out useful, a further idea for improvement along similar lines, could be to use splines along the time axis in order to make use of the contineous nature of air pressure at sea level.

References

[1] P. Øgland, *Eksperimentering med enkel kvalitetskontroll og interpolasjon av værstasjonsdata med varierende tidsoppløsning.* KLIBAS-report no. 18/94, DNMI, Oslo, 1994.

[2] P. Øgland, *Automatic interpolation of SYNOP weather observations: Interpol1 and Interpol2.* KLIBAS-report no. 61/97, DNMI, Oslo, 1997.

[3] P. Øgland, *Evaluation of a meteorological radial interpolation method by statistical experiments.* KLIMA-report no. 23/97, DNMI, Oslo, 1997.

[4] P. Øgland, *Evaluation of a double exponential correlation weighted interpolation method by statistical experiments.* KLIMA-report no. 25/97, DNMI, Oslo, 1997.

[5] P. Øgland, *Quality control computer program S-T-F v.2.1: Automatic removal of "bad" INTERPOL2 values.* KLIBAS-report no. 17/98, DNMI, Oslo, 1998.

[6] P. Øgland and Z. Toporowski, *Interpolating air pressure in TELE with INTERPOL_P0.* KLIBAS-report no. 46/98, DNMI, Oslo, 1998.

[7] P. Øgland and Z. Toporowski, *Interpolating air pressure in TELE with INTERPOL_P0 v.1.1.* KLIBAS-report no. 52/98, DNMI, Oslo, 1998.

[8] P. Øgland, *SYNO_KONTR v.2.1: Daily interpolation and quality control.* KLIBAS-report no. 06/99, DNMI, Oslo, 1999.

[9] P. Øgland, *Interpolating in TELE with Hirlam10 forecast data.* KLIBAS-report no. 08/99, DNMI, Oslo, 1999.

What to do at DNMI when the VIND_REG program in KLIBAS breaks down with a message about being unable to open datafiles

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY

(subm. February 17th, 1999)

ABSTRACT

The VIND_REG data transport program is responsible for reading automatic wind registrations from data files and inserting these into the XVIND Oracle data table in the KLIBAS database system. This paper is concerned with discussing a particular problem that makes the system break down when it is unable to read data files, and formulating a solution to the problem.

The VIND_REG program

Specifications for the VIND_REG program were written by Lars Andresen in September 1997, resulting in a first version of VIND_REG being released the 12th of July 1998 [1]. The purpose of the program was to read files, interpret and modify observations when necessary and store the results in suitable datatables in the KLIBAS database.

By August 18th 1998 a routine was being established at the Climatology Division for handling the observations. A quality control program VINDDEK was then released according to specifications by Lars Andresen and P.O. Kjensli [2].

By the end of August the same year, a revised version of VIND_REG appeared [3]. Minor faults in the algorithm had been corrected, the specification had been slightly updated.

Still there were problems with the program, and on the 20th of October 1998 a version 1.2 of VIND_REG was released [4]. The program was now running systemaically every hour of the day by the crontab schedule on the UNIX machine. A new table XVIND was designed for storing the observations.

The first log of the program is in April 1998. It was then run 17 times, failing 0 of these. The curve below shows the relative number of runs of the program that has been failing according to the log.

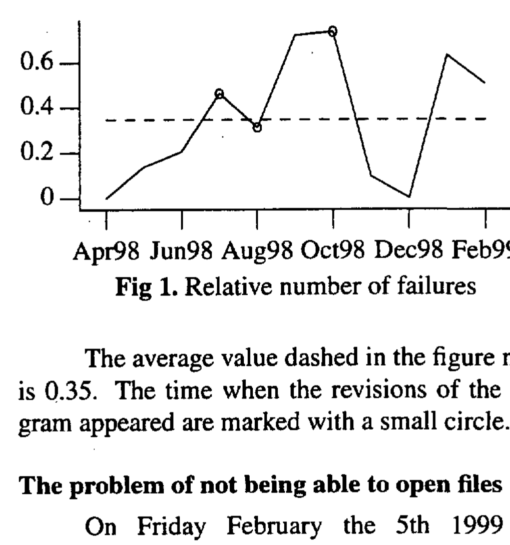


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.35. The time when the revisions of the program appeared are marked with a small circle.

The problem of not being able to open files

On Friday February the 5th 1999 the VIND_REG program collapsed in an attempt to open the file enfg0798.txt which should contain data from Fagernes Lufthavn. The program has then collapsed on almost every run since, at least when no new stations have been added til the list of files which would proceed enfg0798.txt when the files are sorted.

The reason why the program breaks down becomes quite clear when having a look at the directory ~kabase/vindreg, as there is no file by the name of enfg0798.txt here, and consequently it is impossible for VIND_REG to read data from such a file.

The question is, then, why should the program attempt at reading data from this file? The answer is probably that a file by this name has previously been stored on this particular directory, but is now removed for whatever reason.

Why search for ENFG in the first place?

The definition and order of files to be handled by VIND_REG are administred locally by the program on two files, named vind_reg.lst and vind_reg.lst.old. The formula used by the script for selecting current station is first by updating the vind_reg.lst file *ls -1 | sort > vind_reg.lst* and then defining a variable.I file0 by *set file0='diff vind_reg.lst vind_reg.lst.old| sed -n 2p| cut -c2-'* which means that the first item not on both the files vind_reg.lst and vind_reg.lst.old is definded in the variable *file0* and used for running the program.

When VIND_REG has run successfully for the file *file0* then this particular file will be added to the vind_reg.lst.old file.

It should be noticed that never is the file vind_reg.lst.old updated in any other way, meaning that once a file has been defined in the file0 variable, VIND_REG will not be pleased until this file has been read by the program, even if the file is later removed from the ~kabase/vindreg directory.

How to solve the problem

The error of the program has to do with the list of files suggested to the program to be read and the files available do not correspond. It therefore seems natural to enlarge the VIND_REG program by having some sort of update of the vind_reg.lst.old file, removing items on the file that no longer correspond to actual files on ~kabase/vindreg, or at least preventing the variable *file0* having such files in its domain.

The simplest way to make sure that only available files are being considered is perhaps rewriting the definition of the *file0* variable into something like *diff vind_reg.lst vind_reg.lst.old| grep "<" | sed -n 1p| cut -c2-* where the 'grep "v"' part would then secure that only files on the newly updated vind_reg.lst, not being on the older vind_reg.lst.old, shall be considered. The old vind_reg.lst.old will then grow on forever, containing both the name of the files still available on ~kabase/vindreg and the name of files not available.

This should solve the current problem, although it would not help solve problems having to do with files being altered. The version 1.2 of VIND_REG only reads the first version of a particular datafile, but this is a problem which would have to be handled in a more subtle manner, probably in a next version of the program.

References

[1] P. Øgland, *Computer program VIND_REG.* KLIBAS-report no. 22/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program VINDDEK.* KLIBAS-report no. 39/98, DNMI, Oslo, 1998.

[3] P. Øgland, *Computer program VIND_REG v.1.1.* KLIBAS-report no. 43/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

A problem with reading cloud cover N for PIO observations into the KLIBAS database system at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY

(subm. February 17th, 1999)

ABSTRACT

As the PIO_INN program is responsible for reading both observations from "PC i observasjonstjenesten" (PIO) and semi-automatic weather stations (SAWS), some special considerations have to be taken, especially for the SAWS stations, as some of the codes used for indicating missing values is less than fortunate, sometimes using the value 0 both as a real value and as an indicator. In this paper a problem relating to cloud cover N is analysed, suggesting a momentarily solution.

The population of PIO weather stations

Medio February 1999, the PIO_INN computer program is reading observations from 11 PIO weather stations. Figure 1 shows the number of files read each month since the intiation of the system in Mars 1998.

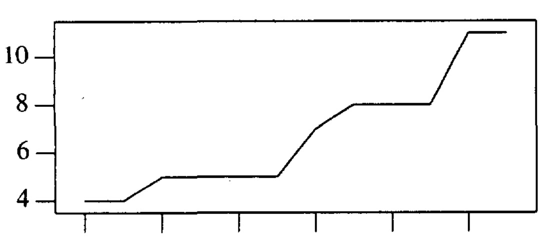


Fig 1. Number of files with PIO observations

On the area where the PIO files are to be found, observation files from semi-automatic weather stations (SAWS) are also placed. The first test SAWS being from June 1998. In July there were 9 stations of this kind, in August 1998 to February 1999 there have been 14 stations.

PIO_INN program development

The development of the PIO_INN system has so far consisted of two phases. Shortly after PIO data were a part of the DNMI dataflow systems on Mars 23rd 1998, see user guide in [1], an initial verson of the PIO_INN program, described in [2], was operative in the sense that it was reading observations from the pio-files into the PIO datatable in the Oracle RDBS of the KLIBAS database system on a daily basis. An instruction on how the Climatology Division were to handle PIO observations is described in [3].

The files generated during these early stages of the PIO project contained only parametres for barometre temperature (Bp), air pressure (P0, PF, PT), evaporation (EV), air temperature (TT, TnT, TN_12, TxT, TX_12), ground level temperature (TG_12), water temperature (TW) and relative humidity (UU), all acronyms explained in [1].

The initial version of PIO_INN was revised in July 1998 by adding log functions in order to add a systematic control on whether the program was running according to specifications or not and to which extent observations were being made at correct time. The version 1.1 of the system, that is described in [4], was also augmented in order to handle a complete set of parametres, as described in [1], not only the test parametres being used on early files.

During August 1998 the PIO_INN system was significantly reprogrammed in order to merge data from semi-automatic weather stations (SAWS) into the PIO dataflow as documented in [5]. Due to the complex nature of the SAWS files, using a mixed approach for marking missing values, including attainable values to signify missing ones, this second version of the PIO_INN system has been so far revised twice, documented in [6] and [7].

In the version 2.1 of PIO_INN, described in [6], statistical charts were added to be produced by the program in order to make it easier to see whether the program was performing normally or not. Problems having to do with data format on the SAWS files was systematically documented and reported to those responsible for the producing the files.

The version 2.2, described in [7], went a step further by adding a simple quality control routine to the program in order to eliminate observations that could not be inserted into the PIO datatable, causing the system to break down, and displaying quality control results by methods of statistical process control (SPC) as a help to detect whether the program was under control or not.

Robustness

The first version of PIO_INN started running on the 12th of May 1998. In figure 2 the relative number of abnormal (defective) terminations of the program so far is plotted.

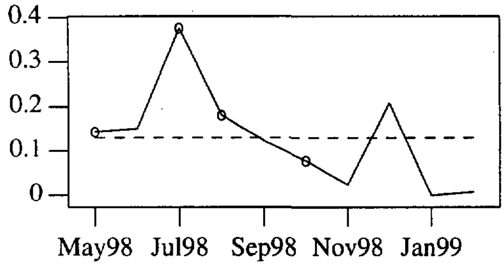


Fig 2. Relative number of executions to fail

As can be seen in figure 2, with exception of December 1998, the tendency since the design of version 1.1 in July 1998 has been a gradual improvement of the system in terms of reporting less and less defects on the average each month. The general average relative number of defects for the eight month period is 0.13 as indicated by the dashed line. The instances when the program was revised are indicated by small circles on the curve.

The program is run about 200 times each month on the average. According to automatic runs by the crontab schedule where it is to be executed every 3 hours or eight times a day, it should run 240 times on a 30 day month. The average being below this value is explained, however, by the three hour schedule run by the crontab prior to redefinitions in October. In the beginning the program was automatically run only four times a day.

The cloud cover problem

The problem of N being given the value NULL (no value) when it should have contained the value 0 was reported in an e-mail by Margareth Moe on January 25th 1999, and once again on February the 8th 1999, giving Lyngdal 28.01.99 at 15:00 UTC and 29.01.99 at 12:00 UTC as examples.

The problem of the cloud cover parameter N containing NULL when it should instead have contained the value 0 has previously been addressed and presumed solved on November 6th 1998, see [8] page 19.

Selecting observations from PIO from station no. 28800 Lyngdal verifies indeed exactly what was reported. When checking the actual code in pio_inn.pc, it seems that even though special treatment is designed for cloud parametres h, Nh, Ns1, C1, hshs1, Cl, Cm and Ch, removing the content of these parametres if they contain the value 0 and are NOT defined as PIO stations, the cloud cover parameter N is not treated in the same manner.

In fact, cloud cover N is assigned the value NULL whenever it is reported to have the value 0, no matter what kind of station it is. This is also verified on the *pio_inn.tst.02* file which contains a catalogue of all the updates done by the program during February 1999. So far a total of 538 assignments of N:=NULL were done according to this file, many of them for non-SAWS stations.

Conclusion and suggestion

As the assignment N:=NULL should only apply to SAWS, not PIO stations, this particular algorithm should be moved from its present position in the "testValues" procedure, and placed along with the special assigments that are done for the non-PIO stations only.

References

[1] O. Bonlid, "Inntastingsprogram for Synop-stasjoner. Brukerveiledning for Pio versjon 3.2," in *PC i observasjonstjenesten: PIO_INN v.1.1.* (appendix), KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program PIO_INN.* KLIBAS-report no. 19/98, DNMI, Oslo, 1998.

[3] Å.M. Vidal and P.O. Kjensli, "5 PIO stasjoner ble satt i drift 24. mars" in *PC i observasjonstjenesten: PIO_INN v.1.1.* (appendix), KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[4] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.1.1.* KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[5] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.0.* KLIBAS-report no. 41/98, DNMI, Oslo, 1998.

[6] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.1.* KLIBAS-report no. 53/98, DNMI, Oslo, 1998.

[7] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.2.* KLIBAS-report no. 58/98, DNMI, Oslo, 1998.

[8] P. Øgland, *KLIBAS Process Improvement November 1998.* KLIBAS-note no. 26/98, DNMI, Oslo, 1998.

A procedure for renumbering automatic weather stations in KLIBAS with consequences for the ALN data table

Petter Øgland
Per Ove Kjensli

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo

(subm. February 18th, 1999)

ABSTRACT

The purpose of this paper is to document the procedure used at DNMI when altering station number for the automatic weather stations. The procedure is discussed as a part of the data collection systems of KLIBAS (DATAINN) and is analysed in this context. Renumbering for seven stations are used to give detailed examples, illustrating different aspects related to renumbering.

The DATAINN program

DATAINN is an undocumented KLIBAS program that is designed for analysis and quality assurance of the complete set of data collection programs used by the system, based on specifications in [1].

There are currently 7 data collection routines defined to be supervised by DATAINN. The most important of these is the synop data collection [2], which has been revised several times either for debugging of performance optimalisation.

The AUTO_INN system, collecting hourly observations from automatic weather stations [3], is another very central part of KLIBAS. The current version was last revised on January 5th, 1999.

We then have the PIO_INN program, reading observations from PIO stations and semi-automatic weather stations into the PIO data table [4]. This program was last officially revised in October 1998.

Another important routine is the META_INN program [5], reading metar weather observations from a similar format as the synop observations. The last version of this program dates also from October 1998.

The program VIND_REG is responsible for collecting wind observations from files, the so-called AERO routine which was specified during the autumn of 1997 and established in its present preliminary version during 1998 [6].

Finally, the AANDERAA program [7], is the last of the systems being monitored by DATAINN, and although this particular program has not been revised since February 1995, it is still one of the more important routines in the system.

Renumbering stations

In the present version of the KLIBAS database system, the number used for identifying a particular station plays a vital role, as data tables use this number as a part of their name [8]. For various reasons, however, sometimes it is necessary to renumber stations. This has been done on several occations recently with the automatic weather stations.

Presently there are seven AWS stations that need to be redefined in order to update information in ALN, updates in ALN necessary before establishing precipitation stations in the new dataware house system that is a part of the 1999 redefinition of the KLIBAS database system. The need for this update was logged by the DATAINN system from October the 30th and onwards, added to the DRIFT warning system [9] on January 27th 1999.

Below, statistics from each of the seven updates are described, starting with the lowest number and working upwards.

(1) ØSTRE TOTEN - APELSVOLL

Data stored on table A11510 should be moved to table A11500. When checking this station, the update seems to have been already done. There is no data table by the name of A11510, but the table A11500 contains 35153 rows. The station contains data from January 1994 to February 1999, although there are gaps in the data set, August to October 1994 is missing and October 1995 to April 1996 is missing.

According to the datatable ST_TYPE, Apelsvoll was defined as an ordinary climatological weather station from 1930 to 1987. It was the replaced by an EDAS auotmatic weather station, which was replaced by a Campbell AWS in 1993.

On the directory ~kabase/automat/data there are backupfiles for 11510 for 1995 and 1996. By renaming these files by substituting the number 11510 with 11500 it is now possible to run the MND2ALA program [10], which then inserts the observations for the second of the two gaps described above.

In order to find out if the gap from 1994 can be filled in the same way, the file *manadsfiler.94.tar.gz* needs to be unzipped and untared. A safe way of doing this is to start with the command *cp manadsfiler.94.tar.gz manadsfiler.94.tar.gz.old,* making sure that the original file is not destroyed, the un-zipping the file by *gunzip manadsfiler.94.tar.gz* and then untar the files by *tar -xpf manadsfiler.94.tar* which produces 317 files.

By listing the files **11510** we see that there are twelve of these, and the files *08011510.c94, 09011510.c94* and *10011510.c94* seemingly containing data. The procedure described for filling the first gap, then, is then used for filling this gap.

(2) KISE PA HEDMARK

Data stored on table A12560 should be moved to table A12550. When checking this station, by use of the program AARSREKKER-AMME [11], it appears that also in this case the update has been done, although for Kise both the tables A12560 and A12550 exist, although A12560 has not been updated since October 1998.

The table A12560 is thus removed from the database.

(3) LØKEN I VOLBU

Data stored on table A23510 should be moved to table A23500. For this station the situation is exactly the same as for Apelsvoll, and the two gaps are filled in similar fashion.

Similarily to Kise, both the tables A23510 and A23500 contain data, although A23510 only contain observations from 1997 and 1998, not being updated for 1999. Just like was done for Kise, also here we drop the table A23510.

(4) LANDVIK

Here we have a slight modification of the previous situations. Observations should be moved from table A38150 to A38140. In this case the table A38140 is not available from the casual KAXX user, but the table does, in fact exist, although only visible to the AWS system user 'A', and the table is up to date with the exception of a gap in 1994, similar to previous situations, and a gap in July to September 1996 that may perhaps have something to do with the station rather than a flaw in the data collection system.

The first thing to do for this station is to establish a public synonym for the table A38140 so that it may be accessible for all users of the KLIBAS database system. This is done by the SQL statement *create public synonym a38140 for a.a38140* executed by the AWS system user A. The table a38150 should the be removed, and when this is done, the backup files for 1994 and 1996 should be read as in the previous cases. When this was done, both the gaps from 1994 and 1996 were filled.

(5) FURENESET

Observations should be moved from table A56430 to A56420. Just like the previous station, the table A56420 was not generally available, but it, nevertheless, existed and was even complete, so none of the previous updates by backup files were necessary in this case.

(6) TJØTTA

In this case the situation is different, as no previous update has been done, although both the old table A76540 and the new table A76530 exist. In the old table there are gaps, however, so the easiest way to perform the transaction may be to read backup files directly into A76530. In this case all backup files from 1994 to 1999 have to be read. Observations from 1997 to 1999 can be read from the normal /opdata/automat/data directory by copying these to the ~kabase/automat/data directory using the new s tation number as part of the file name.

The way this kind of problem has been handled in the past, has been by running the MVTABLE program [13] were the transaction from A76540 to A76530 can be defined. In this case, however, it seems easier just to read backup files.

(7) BODØ - VÅGØNES

The final case is similar to the earlier cases. Data should be transferred from A82230 to A82260, but this has already been partially done. There are two gaps, just like for Kise and other stations, one in 1994 and one in 1995/96. The gaps are filled by reading backup files.

The old data table A82230 have to be dropped, although the table is not accessible for ordinary KLIBAS users.

References

[1] P. Øgland, *Data inn - spesifikasjonsrapport.* KLIBAS-report no. 01/94, DNMI, Oslo, 1994.

[2] P. Øgland, *Reading data from syno-files into KLIBAS: SYNO_INN v.3.7.* KLIBAS-report no. 05/99, DNMI, Oslo, 1999.

[3] P. Øgland, *Data transport and storage by program AUTO_INN v.1.2.* KLIBAS-report no. 01/99, DNMI, Oslo, 1999.

[4] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.2.* KLIBAS-report no. 58/98, DNMI, Oslo, 1998.

[5] P. Øgland, *Reading METAR from meta-files into ALF by META_INN v.1.1.* KLIBAS-report no. 57/98, DNMI, Oslo, 1998.

[6] P. Øgland, *Storing AWS wind observations by use of VIND_REG v.1.2.* KLIBAS-report no. 56/98, DNMI, Oslo, 1998.

[7] P. Øgland, *Innlasting av Aanderaa-data til arbeidslager på Typhoon.* KLIBAS-report no. 09/95, DNMI, Oslo, 1995.

[8] P. Øgland et al., *Utarbeiding og testing av ulike datastruktur på Typhoon. Delprosjekt 2.* KLIBAS-report no. 42/92, DNMI, Oslo, 1992.

[9] P. Øgland, *Towards an ISO-9000 standard for system development by DRIFT.* KLIBAS-report no. 47/98, DNMI, Oslo, 1998.

[10] P. Øgland, *Reading AUTO-files into ALA by MND2ALA v.1.2.* KLIBAS-report no. 68/98, DNMI, Oslo, 1998.

[11] P. Øgland, *Utskriftsrammer for verifikasjon og testutskrift av AVS-data: Døgnekstremramme, årsramme, månedsramme, årsrekke-ramme og to-parameter-ramme.* KLIBAS-report no. 16/96, DNMI, Oslo, 1996.

[12] E. Waage, *Brukarrettleiing AUTO. Datainnsamling frå automatiske vêrstasjonar.* KLIBAS-report no. 31/94, DNMI, Oslo, 1994.

[13] P. Øgland, *Computer program MVTABLE v.2.1.* KLIBAS-report no. 30/98, DNMI, Oslo, 1998.

Problems with the AWS at Finsevatn causing it to top the error statistics for CHECK_H_STAT in the TELE quality control routine

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo

(subm. February 22nd, 1999)

ABSTRACT

The TELE data table is used for several climatological purposes. For some programs, such as the KA_H_STAT program, used for producing monthly climatology statistics, one missing value of minimum temperature may cause missing values for several derivated parametres such as TM, AV, TNM, TN, TNDT, TN<0, TN<-10. In this paper one such problem registered at the automatic weather stations (AWS) at Finsevatn is analysed.

The CHECK_H_STAT program

In January 1996 a first description of the SYNOP/TELE routine was published [1]. The purpose of the publication was to describe the system as it was performing primo 1996 in order to use this as a benchmark for further testing. An improved description was published three months later [2].

The first version of the CHECK_H_STAT program was released in May 1998 [3]. The purpose of the program was to check whether the computer program KA_H_STAT was generating reasonable statistcs or not when processing the TELE data table. The idea of the program was to simulate what was already being done manually.

Shortly after, July 1998, a similar approach was made for checking output from the STATUT program [4]. As this was an even more important program, and update vc.1.1 of CHECK_STATUT was released as soon as the next month [5].

The first log of CHECK_H_STAT was in Mars 1998. It was then run 42 times, failing 1 of these (2.4%). The curve below shows the relative number of runs of the program that has been failing according to the log.

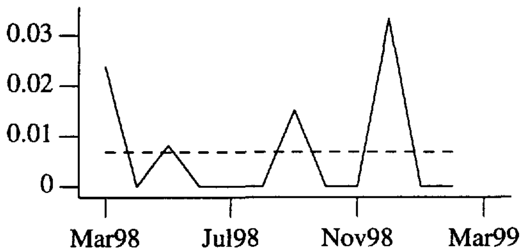


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.0067. There is no apparent visual indication in the figure that might indicate that the program is stabilising although the two last months show a defect ratio significantly below average.

The CHECK_H_STAT results

The plot shows the day by day measurement of defects for the KA_H_STAT output of the month measured by counting the number of missing values in the output file as it is run.

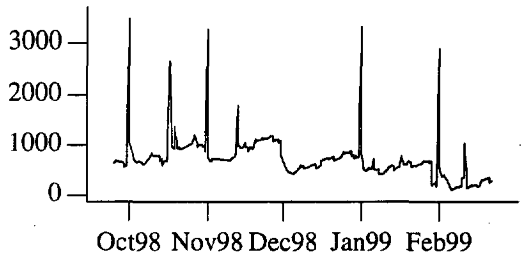


Fig 2. Defects in KA_H_STAT output

The first measurements up to late January 1999 are based on all stations in the TELE data table. As only a selection of these are actually used when producing KA_H_STAT statistics, only stations defined on the TELE_STASJ file have been used since.

Status for present "worst case" station: 25830 FINSEVATN

1. Missing temperature columns, TM, AV, TNM, TN, TNDT, TN<0, TN<-10.
2. Missing precipitation columns, RR, RRPRO, RX, RXDT, RR>=0.1, RR>=1.0, RR>=10.0.
3. Miscellaneous missing columns, KL, OV, BI.

Below we deal with each of the three groups of problems seperately, beginning with the temperature problem.

Temperature on Finsevatn in February

The "worst case" statistics above show that KA_H_STAT was unable to produce several temperature statistics for February. Looking at the TELE data table, we see that the only temperature value missing for February so far is a minimum temperature on the 7th at 18:00 UTC.

This particular value has been removed by the S-T-F program [6] as we notice that the missing value is flagged by the number '5'. When we then check the ALA data table for February the 6th and the 7th, we notice that there are only 46 observations, observations on the 7th at 10:00 and 11:00 UTC missing.

This will then explain why the program ALA2TELE [7] has not been able to produce a minimum temperature, there simply wasn't enough data. This means then that both TN and TX on the 7th at 18:00 UTC were produced by the INTERPOL2 program [8], and as the temperature interpolation apparently was in conflict with consstency checks in S-T-F, the value was removed and flagged '5' according to specifications.

In order to solve the problem, we obviously need a better method of estimation than the one used in INTERPOL2. One solution, that seems quite reasonable at the moment, is to fill in the blanks in ALA by spline interpolation as hourly temperature observations should behave fairly close to lying on a continuous curve.

Precipitation on Finsevatn in February

The "worst case" statistics also show that KA_H_STAT has been unable to produce precipitation statistics for Finsevatn this month. Checking the TELE data table in this case, we also se that the only observation missing is the one on the 7th at 18:00, apparently not having been produced due to lack of data in the ALA data table.

In this case, however, there is no flagg '5' indicating that the observation has been removed by the S-T-F program, so the question is then if any attempt of interpolating precipitation RR was acutally done by INTERPOL2. When selecting rows in TELE for station no. 25830 Finsevatn where FLRR = '4', no rows are selcted, while FLRR = '4' does occur on 2077 other observations in TELE for 1999.

When investigating the definition data table TELE_PARA for stnr no. 25830, we see that the parameter RR is defined here, so according to the internal checks of INTERPOL2, interpolation should be performed, and checking the interpol2.doc file, we see that at least interpolation has been considered as there are estimates on how good such an interpolation would have been.

In fact, the conclusion is not bad at all in table 1 using abbreviations as [9]. All values nicely within bounds of what would naturally be accepted by the INTERPOL2 interpolation program as fair estimates. The program would indeed accept anything with an average error less that 1000 and error standard deviation less than 1000, so almost anything would be accepted anyway.

It is also confirmed on the interpol2.log.02 file that RR has been estimated for Finsevatn. On the 22nd, for instance, RMSE(RR) for Finsevatn ranks as medium quality RMSE(RR) for all stations considered, ranking between RMSE(RR)=0.0 and RMSE(RR)=8.0.

Station name	No.	Sample	Bias	Stde	Rmse	Mae	Emin	Emax
39. FINSEVATN	A25830	10	0.0	0.8	0.7	0.6	0.1	1.2

Table 1. Interpolation validation from INTERPOL2.

Could the explanation be that S-T-F does not insert the flag '5' when removing unacceptable precipitation values? When we check the TELE datatable for rows containing FLRR = '5' there are no rows returned.

When checking the source code for S-T-F, however, there is nothing that seems to indicate this. On the contrary, the update of RR=NULL and FLRR='5' seems to have been implemented in exactly the same manner as with TT.

The only conclusion that can be drawn from this is that a detailed study of the behaviour of the INTERPOL2 program needs to be done in the case of Finsevatn.

Weather on Finsevatn in February

The third item on the list of problems with Finsevatn mentions parametres KL, OV and BI. According to definitions on the final product [10], KL means number of days with clear weather, OV means number of days with overcast and BI means number of days with variable weather.

As Finsevatn is an automatic weather station, none of these are well defined. Consequently the CHECK_H_STAT program should be altered so that it only counts blanks in these columns if the stations observes weather (ww).

Conclusion

The problem with temperature should be solved by interpolation temperature in ALA, for example by use of splines. The problem with precipitation is, however, not solved, and needs more detailed investigation of the INTERPOL2 program. On the other hand, the problem with counting days with special type of weather is not a problem and should be eliminated from the CHECK_H_STAT program.

References

[1] P. Øgland, *Beskrivelse av synop-rutine.* KLIBAS-report no. 02/96, DNMI, Oslo, 1996.

[2] P. Øgland, *Beskrivelse av synop-rutine. Versjon 1.1.* KLIBAS-report no. 09/96, DNMI, Oslo, 1996.

[3] P. Øgland, *Computer program CHECK_H_STAT for monitoring missing values on KA_H_STAT.* KLIBAS-report no. 18/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Computer program CHECK_STATUT.* KLIBAS-report no. 23/98, DNMI, Oslo, 1998.

[5] P. Øgland, *Computer program CHECK_STATUT v.1.1.* KLIBAS-report no. 35/98, DNMI, Oslo, 1998.

[6] P. Øgland, *Quality control computer program S-T-F v.2.1: Automatic removal of "bad" INTERPOL2 values.* KLIBAS-report no. 17/98, DNMI, Oslo, 1998.

[7] P. Øgland, *Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ala, mnd2hla, ala2tele, adk og mkk.* KLIBAS-report no. 59/97, DNMI, Oslo, 1997.

[8] P. Øgland, *Automatic interpolation of SYNOP weather observations: Interpol1 and Interpol2.* KLIBAS-report no. 61/97, DNMI, Oslo, 1997.

[9] P. Øgland, *Evaluation of a double exponential correlation weighted interpolation method by statistical experiments.* KLIMA-report no. 25/97, DNMI, Oslo, 1997.

[10] DNMI *Klimatologisk hurtigoversikt Desember 1998.* ISSN 0808-3681, DNMI/KLIMA, Oslo, 04.01.1999.

Designing a system for automatic monitoring of the ALN data processing routine at DNMI

Petter Øgland
Per Ove Kjensli

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo
(subm. February 23rd, 1999)

ABSTRACT

In order to reach the goal of creating an integrated quality control of precipitation from automatic weather station and traditional weather stations and precipitaiton stations, the ALN precipitation station data processing routine needs to be monitored in order to measure the consequences of integrating these routines. In this paper suggestions for a first version monitoring system of the routine are made.

The PRECIP monitor program

In January 1995 all computer programs made for the ALN data processing routine up until this point were collected in a folder [1]. The folder has since been updated, as the routine has been shaping. A description of the ALN data processing routine was published in January 1996 [2].

In August 1998 a system for collecting statstics in order to automatically monitor the routine was established [3]. The first log of the program was in December 1997. It was then run 12 times, failing 1 of these (8.3%). The curve below shows the relative number of runs of the program that has been failing according to the log.

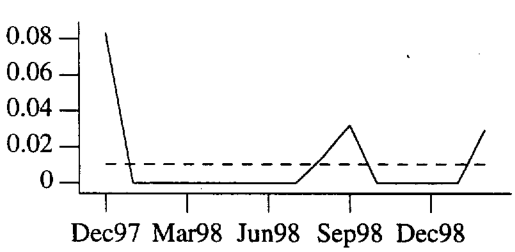


Fig 1. Relative number of failures

The average value dashed in the figure no 1. is 0.01. Apart from the problems in August and September 1998, the program has been running without problems. The dip in February 1999 has to do with reprogramming in order to improve the system.

How and what to monitor

What the current PRECIP v.1.0 program is doing is producing a report that contains statistics from six computer programs working together. Accoring to the programs refered to by PRECIP, the system has been growing according to figure 2.

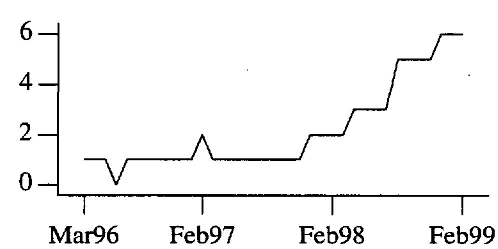


Fig 2. Number of computer programs

The programs chosen to represent the ALN system in figure 2 are the programs that produce log files that make them easy to monitor. Not all the programs with log files are being presently used by the ALN system, and the ALN system contains programs that don't produce log files.

The programs with log files are, however, the following six.

1. *PRECIP.* The purpose of the program PRECIP is to monitor all programs and facilities associated with the ALN datatable (the precipitation routine), predict how these characteristics may develop in the near future and generate warnings or trigger rescue programs whenever predictions indicate problems ahead.

2. *PRECIP_KONTR.* The purpose of the program PRECIP_KONTR is to grow a new version of the PRECIP control routine. Each element in this system is supposed to run independently, but in order to make the system grow before being put to use, the mother program PRECIP_KONTR is supposed to simulate the manual run of the routine.

3. *RRSSU.* The program RRSSU produces a precipitaton matrix for manual areal check. It had to be modified during installation of new Oracle database (v.7.3) during April 1998. The first version of this program is documented in report 03/93 KLIBAS. A revised version 2.0 was then documented in report 20/94. The program RRSSU v.3.0 is mostly revised code from v.2.0 encapsulated in standardised KLIBAS programming code, see [4].

4. *RRUTM.* The program produces precipitation matrices of a different kind than RRSSU. The version 3.0 of RRUTM is functionally similar to version 2.0 (report no. 04/95 KLIBAS), with the exception that the program is also capable of generating statistics from stations defined by lists and is no longer used only for internal purposes, confer [5].

5. *ROMRR* The purpose of the program ROMRR is to help focusing on precipitation observations that might possibly be in error as they deviate significantly from what is observed at stations withing a limited radius. The program automatically updates a list of reference stations when necessary, confer [6].

6. *FREYR.* The purpose of the program FREYR is to implement a prototype quality control system for precipitation data based on the discussion of 1997 of the Nordic FREYR project, cfr. KLIBAS-report no. 04/96 [7].

Important ALN programs that are not part of the PRECIP monitoring system are the data transaction program TELE2ALN and the quality control program GEOK [8]. Not at least these programs are vital to check when monitoring the system, and special attention should be taken in order to provide statistics from these.

In the remains of this paper we will discuss a more detailed plan for design of a monitoring system.

The system map and defect statistics

The most succesful of the monitoring systems so far is the SYNOP monitoring system monitoring the TELE data processing routine [9]. The SYNOP program produces two files, one file *synop.txt* the gives a complete description and status for the TELE dataprocessing system and a file *synop.err.txt* that provides DRIFT with the necessary input for presenting current status of the routine for the monthly report, se [10] page 8.

Exactly the same kind of output should be produced by PRECIP as a general overview of how the computer programs in the ALN routine are performing. This should provide the statistics needed to establish a general overview of the routine.

When such an overview is updated and presented on a daily basis, then special monitoring programs should be made for finding defects in the observations inside the ALN data table. Programs of the type CHECK_H_STAT [11] could be used as a model, producing the type of statistics that are being currently used with great success for the SYNOP routine, confer [10] page 9.

References

[1] P. Øgland and P.O. Kjensli, *Programvare for nedbørrutinen ved Klimaavdelingen.* KLIBAS-report no. 05/95, DNMI, Oslo, 1995.

[2] P.O. Kjensli, *Beskrivelse av nedbørrutinen ved Klimaavdelingen.* KLIBAS-report no. 03/96, DNMI, Oslo, 1996.

[3] P. Øgland, *Collecting PRECIP system statistics.* KLIBAS-report no. 36/98, DNMI, Oslo, 1998.

[4] P. Øgland, *Utlisting av nedbørstasjonsdata fra arbeidslager. Revidert og modularisert programvare.* KLIBAS-report no. 20/94, DNMI, Oslo, 1994.

[5] P. Øgland, *Precipitation matrix program RRUTM v.3.0.* KLIBAS-report no. 12/98, DNMI, Oslo, 1998.

[6] P. Øgland, *Romkontroll for nedbørstasjoner over et selvreviderende referansenettverk.* KLIBAS-report no. 39/94, DNMI, Oslo, 1994.

[7] P. Øgland, *Automatisering av kvalitetskontroll av geofysiske data ved Klimaavdelingen.* KLIBAS-report no. 04/96, DNMI, Oslo, 1996.

[8] P.O. Kjensli, *Geofysisk konstistenskontroll av nedbørdata i arbeidslager.* KLIBAS-report no. 01/95, DNMI, Oslo, 1995.

[9] P. Øgland, *Computer program SYNOP monitoring performance, quality and development of the TELE/SYNOP system.* KLIBAS-report no. 05/98, DNMI, Oslo, 1998.

[10] P. Øgland, *Interpolation and Quality Control status for January 1999.* KLIBAS-note no. 03/99, DNMI, Oslo, 1999.

[11] P. Øgland, *Monitoring KA_H_STAT with CHECK_H_STAT v.1.1.* KLIBAS-report no. 14/98, DNMI, Oslo, 1999.

Testing the Pro*C compiler for Oracle8 on the SGI computer Thunder at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 24th, 1999)

ABSTRACT

In order to migrate the KLIBAS climatological database system from the SGI computer Gale to Thunder and update of the Oracle database from version 7 to version 8, tests have to be applied in order to reduce the risico for failure when the actual migration will take place. This paper addresses the problem of getting the Pro*C compiler to work.

Preparing to migrate the KLIBAS system

In the introduction to the KLIBAS research notes volume one [1], there is a short summary of the development of the KLIBAS climatological database system.

In 1989 it was decided that a new climatological database system should be planned and implemented in order to replace the previous solution. A first version of this, running on a SGI computer called Typhoon and an Oracle6 database, was operative for processing of daily precipitation measurements in 1993 [2].

During the second half of 1994, the Oracle database was updated from version 6 to 7 [3].

The next major update of the database system then occured in 1995, when the Oracle database was updated from version 7.0 to version 7.1 [4].

A database collapse mid April 1998 resulted in updating Oracle to version 7.3 and moving the complete KLIBAS to a new SGI computer galeha. The performance of the database programs improved tremendously as a consequence of this, and earlier problems with system breakdown because of lack of disk space was also solved [5].

A plan for updating Oracle to version 8 and updating the hardware further was also published April 1998 [6]. One important step in reaching this goal is to test the C-compiler and make sure that continued programming will function in the new enviroments, see point seven on the report from meeting the appendix to the KLIBAS statistics from November 1998 [7].

In a note [8] plans for preparing hardware and software updates were described, ending with the conclusion that the Oracle8 database on SGI Thunder is presently not ready for migrating KLIBAS as both the message handler and the Pro*C compiler need to be installed correctly and further testing need to be carried out.

Since then information was sent to responsible persons in order to help out, and alterations and suggestions were made. Below further work on testing the SQL*Plus message handler and the Pro*C compiler is documented.

Using SYNO_INN as a test case

During the previous test [8], the TELE table was not availble, making it impossible to test programs like SYNO_INN. Logging on the Oracle8 on Thunder this time, however, all the vital SYNO_INN tables, such as SYNOP, SYNOP2, TELE and TELE_PARA are available, meaning that there should in principle be no problem running the SYNO_INN program.

When entering the Oracle8 database by the SQL*Plus interface there is no indication, either, that there may be any problems with the message handler. SQL*Plus seems to response in similar manner to how it does on the Oracle7 database on Gale.

Running the SYNO_INN program in its current state, however, does not work, as the paths in the syno_inn.csh script are non-portable, specially designed for Gale, meaning that the first thing to do in order to run the program on Thunder will be to alter the path definitions of the program in order to make it portable. This is done by replacing the pattern /usr/people/kapo with the variable *$HOME.*

Now the program is accessible, but soon breaks down as there is a specific path reference inside the program. The only way to alter this will be by reprogramming that particular bit so the complete program will be portable.

After altering the code and trying to run the program once again, just as with the previous test in [8], the compilation breaks down with the message *cannot get /oracle2/klibas/8.0.4.1/precomp/env_precomp.mk for including.*

After asking people about what to do in this case after the test in early February, it was suggested that the file */oracle2/klibas/8.0.4.1/precomp/demo/proc/demo_proc.mk* should be investigated as this should contain some examples on how to compile.

Below is a sequence of efforts made in order to get the compilator to work.

Step 1: using the demo_proc.mk file directly

As the file */oracle2/klibas/8.0.4.1/precomp/demo/proc/demo_proc.mk* is a make file, similar to the proc.mk file that SYNO_INN depended on for compiling in the Oracle7 environment, the first test consisted of copying demo_proc.mk to proc.mk and run this directly.

The result of this effort was a series of warnings:

1. Building client shared library libclntsh.so ...
2. Call script /oracle2/klibas/8.0.4.1/bin/genclntsh ...
3. /oracle2/klibas/8.0.4.1/bin/genclntsh
5. Cannot create /oracle2/klibas/8.0.4.1/lib/libcommon20536.a - Permission denied
6. Cannot access /oracle2/klibas/8.0.4.1/lib/libcommon20536.a: No such file or directory
7. ld: FATAL 9: I/O error (-lcommon20536): No such file or directory
8. libclntsh.so - Permission denied
9. Built /oracle2/klibas/8.0.4.1/lib/libclntsh.so ... DONE

As a concequence of this, no exe file was generated. The compilation failed, but before going into how and why it failed, we need to check the Pro*C specifications.

Step 2: checking on-line documentation

The file $(ORACLE_HOME)/precomp/doc/proc2/readme.doc contains the on-line documentation for the product Pro*C/C++ 8.0.4. According to the file header, the file contains four sections.

A. Section 1 describes compatibility issues.
B. Section 2 briefly describes the new functionality introduced in this release.
C. Section 3 describes the bugs that are fixed in this release.
D. Section 4 lists known restrictions and bugs in Pro*C/++ 8.0.4. It also provides some usage tips for the new host variable types introduced in this release.

The first section begins with explaining that certain features that were available in the Oracle6 compiler are now no longer available in the Oracle8 compiler. Hopefully this will not effect KLIBAS too much, but that remains to be seen.

Much of the new functionality described in the second section seems to be relecant for the C++ compiler, but as the essential parts of KLIBAS were developed before the C++ compiler was available, all KLIBAS programs are in C rather than C++.

The fourth section contains three paragraphs, the first one on #include vs EXEC SQL INCLUDE, the second one on New Datatypes and the third one on Type Equivalencing Operations. Neither of these seem to be of particular relevance for the problem at hand.

In the appendix, however, it says that "The Pro*C/C++ configuration file precomp/admin/pcscfg.cfg needs to be updated with the appropriate path for $ORACLE_HOME." At the moment this file does not seem to contain anything at all, and has not been updated since the 2nd of May 1998.

Step 3: How and why the compilation failed

Returning to the compilation in step 1, the output from the compilation effort resulted in nine lines, beginning with three lines of information, explaining scripts and shared libraries that were being used.

The problem started when the compiler wanted to create a library file on a restricted area. As it was unable to create this file, a few other error warnings emerged and then the process terminated in error, unable to build the client shared library libclntsh.so.

The question now is whether it was necessary to build this library in the first case. In order to resolve that questions, we look at the demo_proc.mk file, and, in fact, by stripping this file down to just the first line, which is a set up of the make environment by the sentence *include $(ORACLE_HOME)/precomp/lib/env_precomp.mk,* we get the same result.

Obviously it is not possible to do anything with the Pro*C compiler locally before either the sample programs have been compiled and the client shared libraries have been build, or the env_precomp.mk has been modified in some way so that it does not need these libraries.

Conclusion

The Oracle8 database on SGI Thunder is presently not ready for migrating KLIBAS as the Pro*C compiler is not capable of compiling in its present condition. Relevant information must be sent to the responsible persons in order to help clear things up.

References

[1] P. Øgland, *KLIBAS research notes volume 1.* KLIMA-report no. 01/99, DNMI, Oslo, 1999.

[2] M. Moe et al, *Databaseprosjektet i Klimaavdelingen. Status pr. 31.12.1993.* KLIBAS-report no. 03/94, DNMI, Oslo, 1994.

[3] P. Øgland, *Omlegging av databaserutiner ved overgang fra Oracle6 til Oracle7.* KLIBAS-report no. 28/94, DNMI, Oslo, 1994.

[4] P. Øgland et al, *Databasegruppen 1995.* KLIBAS-report no. 01/96, DNMI, Oslo, 1996.

[5] P. Øgland, *Maintenance and Performance Statistics April 1998.* KLIBAS-note no. 10/98, DNMI, Oslo, 1998.

[6] Å.M. Vidal and P.O. Kjensli, *Videreutvikling av DNMIs databaser m/grensesnitt.* Klimaavdelingen, DNMI, Oslo, 01.04.1998.

[7] P. Øgland, *KLIBAS Process Improvement November 1998.* KLIBAS-note no. 26/98, DNMI, Oslo, 1998.

[8] P. Øgland, *Preparing hardware and software update for teh KLIBAS climatological database system at DNMI.* Preprint, DNMI, Oslo, 1999.02.03.

Modifying a couple of quality checks in the ADK quality control system for automatic weather stations at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY

(subm. February 24th, 1999)

ABSTRACT

During the recent months two errors have been noted in the ADK quality control system. The first has to do with checking humidity BT against precipitation RR during winter when BT is not functional and the other has to do with comparing temperature TT and TT#2 which seems not to work under certain circumstances. The papers describes how the updates and changes in the ADK system are done.

The ADK quality control system

The specifications for the ADK system are given in [1]. It was implemented to be run on KLIBAS in December 1994 [2], with later revisions in Mars 1995, version 2.0 [3], July 1997, version 2.1 [4] and October 1998, version 2.2 [5].

The ADK system is designed by modules, using different subprograms to perform different tasks. Adding a new type of checks to the system should then, in principle, not be severely difficult, although the interface for the ADK program need to be specified, i.e. the results produced by the program should fit into the ADK in such a manner that it is compatible with the MKK [6] and AUTO_MKK [7] error statistics programs.

The latest addition to the system was the ADK_REPEAT module, which was added on in January 1999 [8].

The first log of the ADK program is from January 1997. It was then run 18 times, failing 6 of these (33.3%). The curve below shows the relative number of runs of the program that has been failing according to the log.

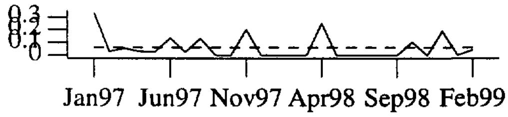


Fig 1. Number of defects

The average value dashed in the figure is 0.06. The curve does not seem to show that the system is stabilising. On the contrary, the system seem to break down rather unsystematically, but the average number of defect runs is low.

Updating the logic test module

Both the BT/RR test and the TT/TT#2 test are located in the logic test module, which is stored on the file *autolog.c* on the directory *~kabase/kontroll/auto/src,* while the binary code, on the other hand, is stored on the directory *~kabase/kontroll/auto/bin* on the file *bautol.*

In order to update ADK one has to enter the *autolog.c* file, modify the BT/RR check so that it only reports error if mnd>5 and mnd<10, remove all tests that involve comparing TT and TT#2, the compile the file using the compile script *make -f autlog.mk* which results in a sequence of statments

1. predecessor cycle (bauto.o)
2. predecessor cycle (logikk.o)
3. predecessor cycle (autlog.o)
4. cc -c autlog.c
5. predecessor cycle (bfunk.o)

6. cc bauto.c logikk.c autlog.o bfunk.o -o bautol
7. bauto.c:
8. logikk.c:

The final result is then a file *bautol* which then needs to be moved to the exe directory *~kabase/kontroll/auto/bin* in order to be run as a part of the ADK system.

Verifying the updates

Data errors of the types discussed above are collected by the ADK system on the directory *~kabase/kontroll/err/logikkfeil* where for example the February 1999 file *02090400.C99* contains 21 "errors" of the BT/RR type. The total number of BT/RR problems in February 1999 is 161, while there are no "errors" involving TT#2.

By simply running the ADK program, and if then the BT/RR problems disappear, then the modification of the system should have been verified. As the modifications of the algorithm having to do with TT#2 was on the same file as the BT/RR it is expected that a verification of the BT/RR results would be a sufficient verification of the TT#2 results as well.

According to log statistics, the program used 750 seconds (11 minutes and 30 seconds) on the average run on January 1999, each run lasting between 274 seconds (4 minutes 34 seconds) and 1621 seconds (27 minutes 1 second). This particular test run lasted 331 seconds (5 minutes and 31 seconds).

According to new check there are now no BT/RR "errors" on the files. The update has been verified.

References

[1] P.O. Kjensli, P. Øgland and S.L. Lystad, *Kontroll av data - spesifikasjonsrapport.* KLIBAS-report no. 01/93, DNMI, Oslo, 1993.

[2] S.L. Lystad and P. Øgland, *ADK/ADL: Datakontroll for EDAS-automatstasjoner.* KLIBAS-report no. 36/94, Oslo, 1994.

[3] P. Øgland, *Implementasjon av datakontroll ADK på Typhoon.* KLIBAS-report no. 35/94, DNMI, Oslo, 1994.

[4] P. Øgland, *Automatisk datakontroll for SYNOP og AUTO: ADK v.2.0.* KLIBAS-report no. 12/95, DNMI, Oslo, 1995.

[5] P. Øgland, *Innlesing AUTO_INN v.1.1 for AVS: Programmer mnd2ala, mnd2hla, ala2tele, adk og mkk.* KLIBAS-report no. 59/97, DNMI, Oslo, 1997.

[6] P. Øgland, *AWS real-time quality control: ADK v.2.2.* KLIBAS-report no. 55/98, DNMI, Oslo, 1998.

[7] P. Øgland, *Monthly AWS Quality Control Statistics MKK v.2.0.* KLIBAS-report no. 49/98, DNMI, Oslo, 1998.

[8] Ø. Nordli and P. Øgland, *Comparison results for the new Glomfjord automatic station versus the present manually run station.* KLIMA-report no. 04/99, DNMI, Oslo, 1999.

A problem with reading snow depth SS and ground description E' for PIO observations into the KLIBAS database system at DNMI

Petter Øgland

Norwegian Meteorological Institute
Niels Henrik Abels vei 40
P.O. Box 43, Blindern
N-0313 Oslo, NORWAY
(subm. February 25th, 1999)

ABSTRACT

A problem has been detected in the PIO data table. In some cases it has been noticed that when SS is equal to 998 and stored as 0, according to definitions, the value of E' is changed. In this paper this problem is discussed and explaind, giving a suggestion to how the problem should be solved.

The PIO_INN program

The development of the PIO_INN system has so far consisted of two phases. Shortly after PIO data were a part of the DNMI dataflow systems on Mars 23rd 1998, see user guide in [1], an initial verson of the PIO_INN program, described in [2], was operative in the sense that it was reading observations from the pio-files into the PIO datatable in the Oracle RDBS of the KLIBAS database system on a daily basis. An instruction on how the Climatology Division were to handle PIO observations is described in [3].

The files generated during these early stages of the PIO project contained only parametres for barometre temperature (Bp), air pressure (P0, PF, PT), evaporation (EV), air temperature (TT, TnT, TN_12, TxT, TX_12), ground level temperature (TG_12), water temperature (TW) and relative humidity (UU), all acronyms explained in [1].

The initial version of PIO_INN was revised in July 1998 by adding log functions in order to add a systematic control on whether the program was running according to specifications or not and to which extent observations were being made at correct time. The version 1.1 of the system, that is described in [4], was also augmented in order to handle a complete set of parametres, as described in [1], not only the test parametres being used on early files.

During August 1998 the PIO_INN system was significantly reprogrammed in order to merge data from semi-automatic weather stations (SAWS) into the PIO dataflow as documented in [5]. Due to the complex nature of the SAWS files, using a mixed approach for marking missing values, including attainable values to signify missing ones, this second version of the PIO_INN system has been so far revised twice, documented in [6] and [7].

In the version 2.1 of PIO_INN, described in [6], statistical charts were added to be produced by the program in order to make it easier to see whether the program was performing normally or not. Problems having to do with data format on the SAWS files was systematically documented and reported to those responsible for the producing the files.

The version 2.2, described in [7], went a step further by adding a simple quality control routine to the program in order to eliminate observations that could not be inserted into the PIO datatable, causing the system to break down, and displaying quality control results by methods of statistical process control (SPC) as a help to detect whether the program was under control or not.

The first version of PIO_INN started running on the 12th of May 1998. In figure 1 the relative number of abnormal (defective) terminations of the program so far is plotted.

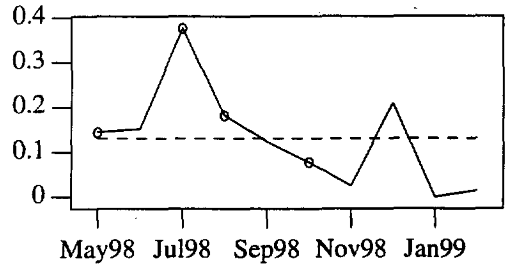


Fig 1. Relative number of executions to fail

As can be seen in figure 1, with exception of December 1998, the tendency since the design of version 1.1 in July 1998 has been a gradual improvement of the system in terms of reporting less and less defects on the average each month. The general average relative number of defects for the eight month period is 0.13 as indicated by the dashed line. The instances when the program was revised are indicated by small circles on the curve.

A general status pr February 1999 for the PIO stations and the PIO system may be found in [8].

The E'/sss problem

What happens with E' when sss=998? According to specifications sss=998 results in the column SS being updated with the value 0, but the column EM is apparently not updated with the content of E'.

In an e-mail on February 8th 1999, Margareth Moe reported this problem, giving Byglandsfjord as an example. For this stations on January 4th, 5th and 6th 1999, E' is equal to 2 on the input file, but the column EM contains the value 3. The corresponding E'sss column on the input file contains the value 2998 in all these cases, and a similar case on the 7th.

In order to find out what is actually happening, the PIO_INN program need to be tested. The program is constructed in such a manner that by setting it in a test mode and defining a few variable to make it read data from 39690 Byglandsfjord for year=1999 and month=1.

When investigating the source code, it appears the problem is due to rounding in the assignment *em=(Esss/1000.0)* that is used inside the function "testValues". Instead of putting the first of the four digits that make up *Esss* into the variable *em*, a decimal number is inserted into the variable, and when this value is attempted inserted into the PIO table, by an SQL insert statment, the value is rounded to the closest integer value.

If *Esss* is equal 2998, *em* is then assigned the value 2.998, which is then rounded up to 3.0 and inserted into the PIO data table.

Conclusion and suggestion

The assignment in the "testValues" procedure should be updated into *em=(int)(Esss/1000.0),* which would the secure that *em* is the truncated integer value that remains after division.

If there are other updates of the kind in the "testValues" procedure, also these should be adjusted in a similar manner, but checking the procedure we see that where such methods have been applied, such as in the cases with cloud observations, and integer trunction similar to the one suggested has already been used.

References

[1] O. Bonlid, "Inntastingsprogram for Synop-stasjoner. Brukerveiledning for Pio versjon 3.2," in *PC i observasjonstjenesten: PIO_INN v.1.1.* (appendix), KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[2] P. Øgland, *Computer program PIO_INN.* KLIBAS-report no. 19/98, DNMI, Oslo, 1998.

[3] Å.M. Vidal and P.O. Kjensli, "5 PIO stasjoner ble satt i drift 24. mars" in *PC i observasjonstjenesten: PIO_INN v.1.1.* (appendix), KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[4] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.1.1.* KLIBAS-report no. 25/98, DNMI, Oslo, 1998.

[5] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.0.* KLIBAS-report no. 41/98, DNMI, Oslo, 1998.

[6] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.1.* KLIBAS-report no. 53/98, DNMI, Oslo, 1998.

[7] P. Øgland, *PC i observasjonstjenesten: PIO_INN v.2.2.* KLIBAS-report no. 58/98, DNMI, Oslo, 1998.

[8] M. Moe, *PIO STASJONER. Status Februar 1999.* KLIMA-report no. 07/99, DNMI, Oslo, 1999.