Free University of Bolzano
Faculty of Computer Science

Optimizing the MicroMet Preprocessor for the Temporal Interpolation of Seasonal Time Series Data

Author
Patrick Clara

Supervisor
Prof. Johann Gamper

Free University of Bozen/Bolzano – Bachelor of Science in Applied Computer Science

March 2012

Acknowledgements

This work and these last three years of study have been possible due to the contribution of many people. Now, I would like to thank them.

First and foremost, it is my great pleasure to express my gratitude to Professor Johann Gamper for having guided me trough the research and for his invaluable support that helped me to overcome the difficulties.

Moreover, I would also like to thank for the motivation and the support that my family gave me during the time of my studies.

Finally, I would like to thank my friends Manuel, Corneliu and Marco for having made my student life such an enjoyable experience.

Patrick

Abstract

A major task in meteorology is to continuously collect measurements about various environmental parameters, such as temperature, rainfall, relative humidity and others. Due to various circumstances, such as broken sensors or connection channels, the impossibility of collecting these values generates data gaps on the time series.

This thesis work analyzes the MicroMet Preprocessor, a theoretical meteorological model developed by Liston and Elder in 2006. It uses various techniques to filter meteorological data, identify deficiencies and anomalies due to the previously described problems, and finally it fills missing data segments according to various temporal interpolation algorithms ranging from basic linear interpolation up to Multiplicative Seasonal Autoregressive Integrated Moving Average statistical forecasting algorithms.

The major contribution of this thesis work is the proposal of several optimizations to the MicroMet Preprocessor to achieve a better accuracy and more realistic results on seasonal time series. Beside this, it includes some extensions to handle various special cases, making the model able to exploit better the known data in the time series and to fill time series with a higher frequency of data gaps.

This thesis work evaluates and compares the improvements from the original MicroMet Preprocessor model by running the implementation of the improved model with real meteorological data over a series of generated data gaps.

Contents

1 Introduction and Problem Description **6**

2 Related Work **9**

3 The MicroMet Model **11**
3.1 Overview 11
3.2 The MicroMet Preprocessor 12
3.2.1 The Data Filtering Step 12
3.2.2 The Data Interpolation Step 13

4 Optimizations of the MicroMet Preprocessor **15**
4.1 Overview 15
4.2 Optimization of the Data Filtering Step 15
4.3 Optimization of the Data Interpolation Step 17
4.3.1 Case 1: Single Missing Time Series Point 17
4.3.2 Case 2: Single Missing Time Series Season 18
4.3.3 Case 3: Multiple Missing Time Series Seasons 21
4.3.4 Efficacy Optimizations 24

5 Implementation **26**

6 Empirical Evaluation **28**
6.1 Methodology and Data Sets 28
6.2 Results 30

7 Conclusion and Future Work **32**

References **34**

List of Figures

1.1 Temporal interpolation example 6
1.2 Data flow diagram of the MicroMet Preprocessor 7

3.1 Rate of Change Limit condition 12
3.2 Case 1 interpolation 13
3.3 Case 2 interpolation 13
3.4 Case 3 interpolation 14

4.1 Spike detection 16
4.2 Special case of the Rate of Change Limit condition 1 16
4.3 Special case of the Rate of Change Limit condition 2 17
4.4 Case 1 interpolation optimization 18
4.5 Case 2 interpolation optimization issue 19
4.6 Case 2 interpolation optimization 1 19
4.7 Case 2 interpolation optimization 2 20
4.8 Case 2 interpolation optimization issue 2 20
4.9 Case 2 interpolation optimization 3 21
4.10 Case 2 interpolation optimization 4 21
4.11 Case 3 interpolation optimization 1 22
4.12 Case 3 interpolation optimization 2 22
4.13 Case 3 interpolation optimization 3 23
4.14 Case 3 interpolation optimization 4 23
4.15 Example of real meteorological data 24
4.16 Efficacy optimizations 25

5.1 Data flow diagram of the application 26
5.2 Screen shot 1 27
5.3 Screen shot 2 27

6.1 Average differences 30
6.2 Frequency distribution 31

1 Introduction and Problem Description

Since the antiquity, meteorology has been concerned with the scientific study of atmospheric conditions, supporting the utopian attempt of the humanity to forecast the weather. Although these studies have been carried on since millennia, less progress has been done until the 18th century. Only the introduction of, at that time, quick communication channels, as the electronic telegraph in the 19th century, allowed the first great discoveries. Not until the development of the computer in the half of the 20th century, the real breakthroughs were achieved. Computers have allowed to process mathematical models integrating the efforts of scientists that have been studying meteorological phenomena and developing physically based mathematical relationships for decades. This have brought to the realization of physically plausible high resolution terrestrial models describing meteorological distribution exploiting the huge amount of data collected by the measurement stations trough the years.

The activity of continuously collecting measurements about various environmental parameters, such as temperature, rainfall, relative humidity and others, have ever been one of the principal tasks in meteorology and the lack of complete data sets can seriously threat the identification of important environmental relationships. Time series with missing values are often encountered in practice due to various circumstances, such as broken of frozen sensors, unreliable communication channels, partial data corruption and others. As an additional drawback, the previously described problems, together with additional environmental influences, can introduce deficiencies and anomalies making the data less reliable. In many cases, measurement stations are situated in localities, which are not easily reachable or also totally unreachable, e.g. during winter time. The failure of such stations can often not be remedied for days or weeks. It is therefore important to develop temporal interpolation techniques, which are able to recreate the missing data in the time series. As Figure 1.1 shows, there are many ways of filling missing data segments with data, but the aim is essentially to create data that reflects best the reality and possibly obeys to known natural laws. The missing data segment is represented by the hatched area. The three light drawn lines conceptually represent different possibilities of temporal interpolations, but only one of them actually seems to be a plausible solution.

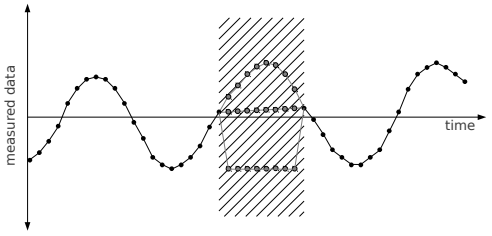


Figure 1.1: Possible temporal interpolations.

MicroMet [7] is one of such previously described meteorological distribution models. It includes a preprocessor responsible for providing complete and continuous time series for the future spatial interpolation, which provides spatially continuous atmospheric data. It filters time series from distinct measurement points searching for anomalies and removing the affected data segments. These segments, together with other preexistent missing data segments, are temporally interpolated by making use of different algorithms in order to fill them with as realistic data as possible. These algorithms range from basic linear interpolation up to complex Multiplicative Seasonal Autoregressive Integrated Moving Average (MSARIMA) statistical algorithms, allowing to interpolate missing data segments of up to several days of length. MicroMet, as also other models do, concentrate the efforts on the spatial interpolation, leaving the possibility for potential optimizations. Figure 1.2 conceptually shows the data flow of the MicroMet Preprocessor.

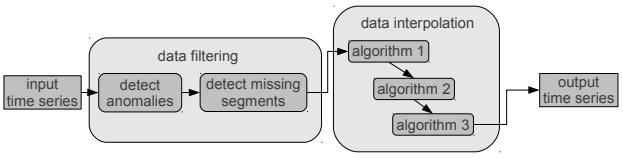


Figure 1.2: Data flow diagram of the MicroMet Preprocessor.

In this thesis, several optimizations of these algorithms are going to be proposed. The optimizations have been thought especially for seasonal non-stationary time series, which present a regular seasonal cycle. The research has actually been focused on air temperature, since air temperature represents one of the most important input parameters in meteorological-physical models and allows to derive or estimate other parameters using natural laws. The optimizations include improvements of the filtering algorithms, which improve the detection of anomalous data segments, as well as improvements of the temporal interpolation algorithms, increasing the plausibility and the reliability of the interpolated data segments. Further, some extensions to these algorithms are proposed, in order to increase flexibility end efficacy. These optimizations have been designed in a way, such that they can be adapted to the time series, allowing them to fit best.

Finally, the implementation of the MicroMet Preprocessor, together with the proposed optimizations, has allowed an empirical evaluation of the improvements and a comparison of both preprocessor versions, proving the actual benefits. Tests have been run over multiple time series collected by different measurement stations, presenting different sizes of missing data segments.

To summarize, the main contributions of this thesis are:

- The optimization of the quality of the interpolated data.
- The optimization of the capability of detecting anomalous data.
- The optimization of the efficacy.
- A tool that implements and optimizes the MicroMet Preprocessor.

The rest of the thesis is organized as follows: Chapter 2 reports research and existing applications related to this thesis. Chapter 3 describes MicroMet and its functionalities. Chapter 4 proposes several optimization to the MicroMet Preprocessor. Chapter 5 shows the most relevant details about the implementation. Chapter 6 presents the results of the evaluation and compares the improvements. Finally, Chapter 7 concludes this thesis and presents possible future improvements.

2 Related Work

The interpolation of missing data segments in time series is actually an old problem, but only recently, with the evolution of powerful computers, able to handle the huge amount of collected data from the measurement stations, it gained popularity under the researchers. During the research that brought to the development of this work, a number of libraries and applications were found that could somehow handle the problem, but also different theories and research outcomes were encountered. This chapter shortly presents some research done until now and the most relevant existing applications, discussing also their limitations and applicability in this specific field.

The interpolation of missing data segments in non-stationary time series, independently from its domain, has been studied by different researchers in the last few decades. So, Brubacher and Wilson [2] were probably one of the first who studied the application of statistical algorithms for estimating missing values in time series. Specifically, they exploited the research made by Box and Jenkins [1] on ARIMA models to estimate the influence of holidays on electricity demand.

Basing on the outcomes of the cited works, other researchers contributed to this methods, like Gomez, Maravall and Peña [15], which presented and compared the usage of both the ARIMA algorithm using the Kalman filter and the EM algorithm for interpolating missing values. As well as Pourahmadi [11] researched in the same field. Later, Cheng and Pourahmadi [3] have contributed to the prediction of time series with incomplete past and to the problem of interpolation of missing values using the autoregression and moving average algorithms. More recently, Weerasinghe [16] discussed some variations and different usages and inputs of ARMA, ARIMA and EM algorithms for temporal interpolation on sulphur dioxide levels.

New upcoming techniques, different from ARMA/ARIMA have been studied in 2010 by Tektas [14], which compares weather forecasting using ANFIS and ARIMA. ANFIS has showed to be a valid alternative to some variants of ARIMA if used in some circumstances. Although these first promising results, its suitability its still heavily discussed an researched.

All this research brought with the time to the development of several applications implementing different algorithms. MeteoIO [9] is a software library that aims to facilitate access to meteorological data over time and space by transparent re-sampling, filtering, temporal and spatial interpolation. The result is presented as grids of interpolated data, which coincide with digital elevation models given as input together with the meteorological time series. From a technical point of view, MeteoIO and MicroMet are similar and coincide in many aspects. A detailed analysis of the MeteoIO source code has actually shown, that some spatial interpolation algorithms from MicroMet have been used. Regarding the temporal interpolation techniques, MeteoIO clearly lacks behind MicroMet. The capabilities of the supported algorithms, with the linear interpolation, the nearest neighbor interpolation and a cumulative algorithm designed specifically to be applied on rainfall measurements, are heavily limited

and not able to provide realistic results in cases of missing data segments.

JGrass [4], together with JGrasstools [5] is an analysis system for geographic resources based on the uDig GIS framework, specifically for management, analysis and modeling of hydrological and geomorphological data. As for MeteoIO, the ability of handling meteorological time series in the correct way is essential. While it concentrates mostly on spatial interpolation, it lacks the ability of getting rid of missing data segments using temporal interpolation techniques.

An other tool, called SEST [13], developed at the University of Zurich, is primarily the outcome of research regarding alignment and similarity of time series. This tool, in comparison to the previously described ones, is not focusing on the application in specific fields, instead, it aims to optimally apply on time series data from a variety of domains. Among others, it has the capability of filling missing values using different algorithms. Beside some trivial ones, it uses the singular value decomposition method to create the main trends in the missing data segments.

Among the software that is able of interpolation, we also have the MatLab [8] numerical computing environment and its open source counterpart GNU Octave [10] as well as GNU R [12]. These applications were actually not intended to interpolate missing segments in time series but in numerical functions. Mainly their ability resides in linear interpolation, nearest neighbor interpolation, spline interpolation, cubic and piece-wise cubic interpolation, which would dramatically fail if compared to the expectations for meteorological data interpolation.

3 The MicroMet Model

3.1 Overview

MicroMet is a nearly physically based meteorological distribution model thought for high spatial resolutions developed by Liston and Elder in 2006 [7]. MicroMet, together with EnBal, SnowPack and SnowTran-3D aggregate to a larger and more complex meteorological model called SnowModel [6]. SnowModel was created with the aim of simulating processes like snow accumulation, blowing snow redistribution and sublimation, snow density evolution and snowpack melt.

The role of MicroMet inside SnowModel is to define the meteorological forcing conditions and their spatial distribution trough assimilation and interpolation, making use of known relationships between meteorological parameters and area topography, in order to provide temporally and spatially continuous atmospheric data. This data is essential to run EnBal, SnowPack, and SnowTran-3D. So, MicroMet represents the first element of the data flow chain inside SnowModel.

The MicroMet model is, from a functional point of view, composed of two main parts:

1. The first part consists of the so called MicroMet Preprocessor. It is responsible for the temporal distribution and the continuity and consistency of measured meteorological data at each measurement point. This is achieved in two steps:

   a) It filters meteorological data on a temporal and on a per measurement point basis and identifies deficiencies and anomalies due to possible defective measurement instruments.

   b) Missing and anomalous data segments are temporally interpolated and filled with as realistic values as possible.

2. The second part is responsible for the spatial distribution of the preprocessed meteorological data. It spatially interpolates the meteorological data of a given time frame over a specific spatial domain. Then, it applies physically based sub-models, dependently from the meteorological parameter. This can be done with the purpose of increasing the data realism or to derive the value of specific meteorological parameters, starting from a combination of other, not unknown, meteorological and/or topographical parameters on a per spatial point basis.

3.2 The MicroMet Preprocessor

The MicroMet Preprocessor is responsible for the consistency and continuity of the meteorological time series data collected for each measurement point, acting on deficiencies, anomalies and missing data segments arisen due to possible defective measurement instruments. This is a prerequisite for applying spatial interpolation algorithms as in the second main functional part of MicroMet.

The MicroMet Preprocessor applies algorithms that filter the meteorological data on a given time frame for a specific measurement point. Found missing and anomalous data segments are afterwards temporally interpolated in order to achieve continuity.

These algorithms are based on the prerequisite that the raw meteorological data collected at the measurement points is provided on a constant time increment. Furthermore, they require the meteorological data to have a seasonal trend, or in a more specific point of view, to have diurnal cycles. The temporal interpolation of missing and anomalous data segments also assumes that the meteorological data of two consecutive seasonal cycles behaves similarly.

3.2.1 The Data Filtering Step

The MicroMet Preprocessor performs three types of quality control tests by looking at the meteorological data in order to find deficiencies and anomalies that met one of the following conditions. Resulting data sections are treated as unreliable and therefore eliminated. The aim is to eliminate not correctly retrieved data due to possibly defective measurement stations.

The Range Limits condition checks the data set for values outside acceptable minimum and maximum limits. Theses values depend from the meteorological parameter the data set is based on and from influencing environmental and topographical parameters.

The No Observed Change within Time Limits condition checks the data set for consecutive constant values over time frames that exceed a maximum length. The maximum length of the time frame do not necessarily depend from any meteorological, environmental or topographical parameter.

The Rate of Change Limits condition checks the data set for consecutive values that exceed a maximum increment limit. The maximum increment depend again from the meteorological parameter the data set is based on and from influencing environmental and topographical parameters. A conceptual representation is shown in Figure 3.1. The data section inside the hatched area is going to be eliminated.

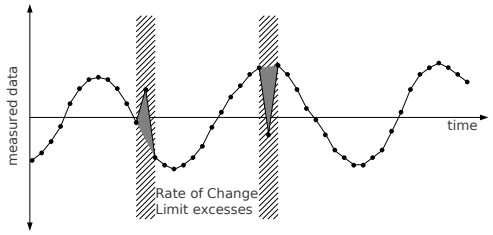


Figure 3.1: Conceptual representation of the Rate of Change Limit condition.

3.2.2 The Data Interpolation Step

The application of the three filtering conditions previously described in chapter 3.2.1 "The Data Filtering Step", together with preexistent missing data segments, results in a time series containing several empty data segments that have to be filled in order to achieve consistency and continuity inside the time series. Different temporal interpolation algorithms are applied distinguishing among three cases, depending on the length of the empty data segments.

Case 1: Single Missing Time Series Point. The first case is met if the empty data segment contains exactly one single time series data point. The missing point is filled by linearly interpolating from the first point preceding the missing one to the first one that succeeds it. A conceptual representation is shown in Figure 3.2. The data section inside the hatched area has been interpolated. The interpolated points are represented as squares, while the points used for the interpolation are represented as rhombuses.

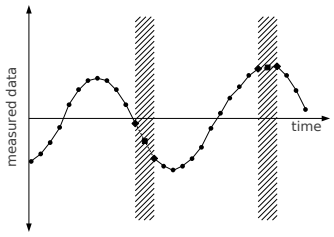


Figure 3.2: Conceptual representation of the Case 1 (single missing time series point) interpolation.

Case 2: Single Missing Time Series Season. The second case is met if the empty data segment contains exactly two or more time series data points and its length is inferior as a single seasonal cycle. Each missing point is filled by linearly interpolating from the point preceding the missing one of exactly one seasonal cycle to the one that succeeds it of exactly one seasonal cycle. A conceptual representation is shown in Figure 3.3. The data section inside the hatched area has been interpolated. The interpolated points are represented as squares. The points used for the interpolation are represented as rhombuses.

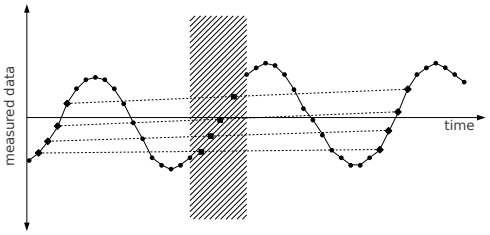


Figure 3.3: Conceptual representation of the Case 2 (single missing time series season) interpolation.

Case 3: Multiple Missing Time Series Seasons. The third case is met if the empty data segment length equals or exceeds a single seasonal cycle. The missing points are filled by making use of the Multiplicative Seasonal Autoregressive Integrated Moving Average statistical algorithm (MSARIMA, often also referred to as ARIMA although MSARIMA is a special form of ARIMA). MSARIMA is used to forecast into the empty data segment by looking at a segment of data preceding the empty one. The length of the segment used for MSARIMA is chosen such that it corresponds to the length of the empty data segment. In an analogous way, MSARIMA is used to backcast into the empty data segment by looking at a segment of data succeeding it. This is resulting in two distinct values for each point in the empty segment. The final result is determined by taking the weighted average of the two values. As weight, the distance in time from the first known point before, respectively after, the empty segment to the point being computed over the whole length of the empty segment is used for the forecasted value, respectively for the backcasted value. A conceptual representation is shown in Figure 3.4. The data section inside the hatched area has been interpolated. The forecasted and the backcasted time series using MSARIMA are represented by a fine dotted line. The final result instead is represented by a thick dotted line. The data segments used for the forecasting and the backcasting are highlighted.

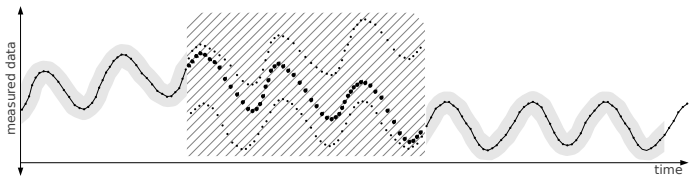


Figure 3.4: Conceptual representation of the Case 3 (multiple missing time series seasons) interpolation.

4 Optimizations of the MicroMet Preprocessor

4.1 Overview

This chapter proposes and discusses several possible optimizations of the original MicroMet Preprocessor. First, it handles optimizations for the data filtering algorithms, which aim to detect a major quantity of anomalies and make it more effective filtering them out. Then, three main optimizations to the data interpolation algorithms are presented, in order to make them generate more reliable and plausible data as possible while filling empty data segments. Finally some minor optimizations are presented, which make the model more flexible and make it better exploit the known data to correct deficiencies in time series with a higher frequency of anomalies and data gaps.

Although Micromet assumes air temperature, relative humidity, wind speed, wind direction and precipitation to have diurnal cycles, these optimizations have been developed with real seasonally behaving meteorological parameters in mind, specifically on air temperature, which is well known for his diurnal variation. An other aspect that suggested and influenced these optimizations is the fact that MicroMet, being a part of SnowModel, is thought for climates and conditions where snow occur, which may behave in a more stable way than different climates.

4.2 Optimization of the Data Filtering Step

The MicroMet Preprocessor, while performing data filtering, applies three distinct methods for detecting unreliable data segments.

While the Range Limit condition and the No Observed Change within Time Limit condition seem to work just fine, there are some issues with the Rate of Change Limit condition, specifically when handling a series of special cases. By simply iterating over the time series and checking for an excess of a predefined rate of change limit between the actual inspected point and the preceding one, the simple elimination of the inspected point without further conditions can lead to the actual elimination of meaningful data while leaving other data, showing enough evidence of being unrealistic, untouched. This holds also if the limit checking is done between the actual inspected point and the succeeding one, eliminating it in case of an excess. The following discussion is based on the first case.

The main issue in handling special cases resides in the ability of determining which of the two points, or even if both, should be eliminated. Furthermore, practice has shown that some

meteorological parameters, as air temperature, can undergo a significantly relative change, causing the need of setting a relative high rate of change limit in order to avoid false positives. It follows that spikes, which usually are created by smaller relative changes, but still show evidence of not being coherent with reality, are not detected.

This suggests the addition of a very basic spike detection algorithm, which checks the rate of change from the inspected point to the preceding point and to the succeeding one using a different, smaller limit. The inspected point should be threated as a spike only if both of the two checked rates exceed the limit and have the same direction. Using two different limits for performing the Rate of Change Limit check depending from the behavior of the data around the inspected point can lead to better detection of unreliable data segments, as conceptually shown in Figure 4.1.

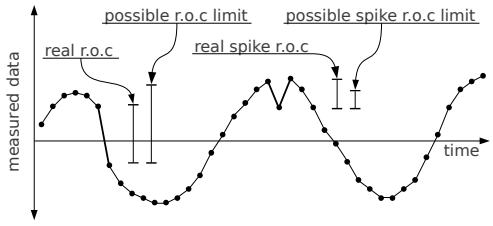
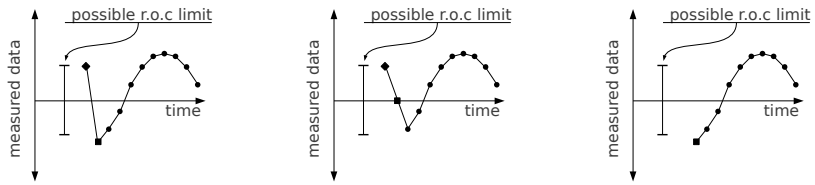


Figure 4.1: Conceptual representation of the spike detection algorithm.

The MicroMet Rate of Change Limit condition shows other difficulties in handling excesses that occur immediately after an empty data section. When the rate of change limit is exceeded between the first point after an empty data segment and the inspected point, which is the second point after the empty data segment, and does not behave like a spike, the inspected point is eliminated leaving the probably unrealistic first point untouched. After letting MicroMet apply the Case 1 (single missing time series point) linear interpolation algorithm, we get a similarly unrealistic situation as before. The conceptual representation in Figure 4.2(a) shows that, by applying the MicroMet Rate of Change Limit condition, the point represented as a square gets eliminated, while the unrealistic point, represented as a rhombus, is leaved unchanged. As shown in Figure 4.2(b), after the point represented as a square gets eliminated, the MicroMet Case 1 (single missing time series point) linear interpolation algorithm creates again an ambiguous situation.

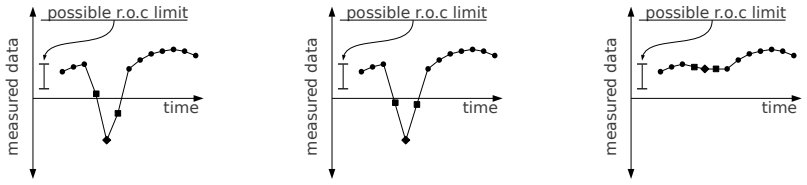


(a) (b) (c)

Figure 4.2: Conceptual representation of handling Rate of Change Limit excesses that occur immediately after an empty data section.

The same issue is encountered in situations where more rate of change limit excesses follow each other. In these cases, not all unrealistic points are eliminated, since every second point is

skipped. Again, by applying the Case 1 (single missing time series point) linear interpolation algorithm, we get a similarly unrealistic situation as before. The conceptual representation in Figure 4.3(a) shows that, by applying the MicroMet Rate of Change Limit condition, the unrealistic points represented as squares get eliminated, while the point, represented as a rhombus, is leaved unchanged, although it should be eliminated too. As shown in Figure 4.3(b), after the points represented as squares get eliminated, the MicroMet Case 1 (single missing time series point) linear interpolation algorithm creates again an ambiguous situation.



(a) (b) (c)

Figure 4.3: Conceptual representation of handling Rate of Change Limit excesses that follow each other.

The two previously described issues can easily be resolved by eliminating the first point after an empty data segment instead of the actual inspected point (i.e. the second point after the empty data segment) in case of a rate of change limit excess. Remember that, as described before, this is only necessary if the inspected point is not a spike, since spikes are already detected by the previously introduced spike detection algorithm, also if occurring immediately after an empty data segment.

The conceptual representation in Figure 4.2(c) shows the correct method of solving the first issue by eliminating the first point and leaving the correct one, represented as a square, unchanged. Figure 4.3(c), instead shows the correct method of solving the second issue by eliminating the points represented as squares and as rhombuses. In this case the interpolation is done using the MicroMet Case 2 (single missing time series season) interpolation algorithm.

In some rare cases, where a data segment, containing exactly two points that exceed the rate of change limit, is found between two empty data segments, either eliminating the first one or the second one can lead to a suboptimal solution. Since the reliability on both points is expected to be low, the elimination of both points is the best solution.

4.3 Optimization of the Data Interpolation Step

The MicroMet Preprocessor applies distinct algorithms by distinguishing among three cases for temporally interpolating empty data segments. Optimizations are proposed for each one of the three cases.

4.3.1 Case 1: Single Missing Time Series Point

The MicroMet Case 1 (single missing time series point) linear interpolation algorithm does actually not show any flaws or practical problems and does also not provide much room for

optimizations. There are, although, two special cases, which it is not able to handle. These occur when an empty data segment containing exactly one point is found at the beginning or/and at the end of the time series, i.e. the first or/and the last points in the time series are missing. In these cases, a linear interpolation between the point preceding the missing one and the point succeeding it is not possible, since either one or the other would lie outside the time series bounds.

The above discussed limitation can be overcome by linear extrapolation, making use of two points, both either preceding or succeeding the missing one. Figure 4.4 shows a conceptual representation of the Case 1 (single missing time series point) interpolation optimization, where the missing segment is found at the beginning of the time series. The data section inside the hatched area has been interpolated. The interpolated point is represented as a square. The points used for the interpolation are represented as rhombuses. The situation, where the missing segment is found at the end of the time series is analogous.

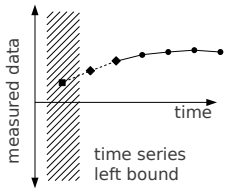


Figure 4.4: Conceptual representation of the Case 1 (single missing time series point) interpolation optimization.

4.3.2 Case 2: Single Missing Time Series Season

The MicroMet Case 2 (single missing time series season) interpolation algorithm comes up with multiple issues, which not only limit the reliability of the interpolated data, but also the ability of actually generating data for the empty data segments in some unusual situations.

Since each missing point is filled by linearly interpolating from the point preceding the missing one of exactly one seasonal cycle to the one that succeeds it of exactly one seasonal cycle, it is possible that one or more of the needed points for the interpolation are missing as well. A simple workaround would be to use points preceding, respectively succeeding, the missing point of two, or is necessary of more, seasonal cycles. Such a solution would although generate less stable and possibly unrealistic data, beside introducing the possibility of generating additional spikes, as represented in Figure 4.5. The hatched areas represent the empty data segments. The interpolated points are represented as squares. The points used for the interpolation are represented as rhombuses. In the example a spike has been generated, which is shown in the shaded area.

A better solution is to require all the points used for the interpolation which precede, respectively succeed, the empty data segment, to be in a single data segment, i.e. following directly each other. This results in a more stable trend and in less or smaller spikes. Before doing the interpolation, the algorithm should check if the two data sections containing the points supposed to be used for the interpolation are continuous, otherwise use sections one more seasonal cycle away. This procedure should be repeated until either two valid segments

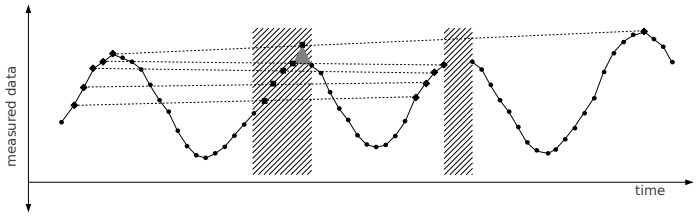


Figure 4.5: Conceptual representation of an issue with a possible optimization.

are found, the bounds of the time series are reached or a predefined limit of iterations are reached. Note that the successful interpolation is possible only if the first case applies. The limit of times the above procedure can be repeated until giving up, should be small as between two and four. Figure 4.6 conceptually represents the proposed solution. The hatched areas represent the empty data segments. The interpolated points are represented as squares. The points used for the interpolation are represented as rhombuses. In comparison to Fugire 4.5 no spikes have been generated anymore, although we still have unsmooth transitions between the interpolated data and the preexistent data.

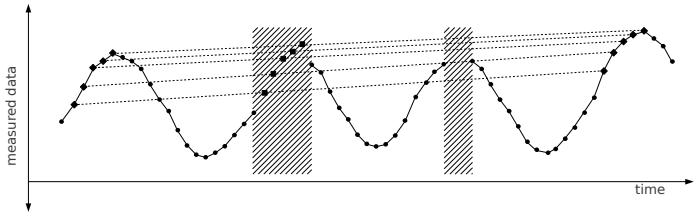


Figure 4.6: Conceptual representation of the optimization.

As previously described, there are two situations where the interpolation is not possible: when, searching for continuous data sections, the bounds of the time series are reached or a predefined limit of iterations are reached. Both situations have in common that either one needed data segment or both have not been found. In the case where only one consistent data segment has been found, either preceding or succeeding the empty data segment, the interpolation can still be done by introducing a similar technique as for the Case 1 (single missing time series point) optimization. Instead of filling the empty data segment by linearly interpolating each point between two points on different sides, it is possible to do it by linearly extrapolating from two points on the same side. Again, as before, if necessary, the data segments used for the interpolation can be more than one seasonal cycle away from the empty data segment, but, in order to achieve better results, they should distance from each other a single seasonal cycle. Figure 4.7 conceptually represents the proposed optimization. The hatched areas represent the empty data segments. The interpolated points are represented as squares. The points used for the interpolation are represented as rhombuses.

The last MicroMet Case 2 (single missing time series season) interpolation algorithm issue has already been seen in Figure 4.6. Independently from the application of the until now discussed optimizations or not, the algorithm is not able to avoid unsmooth transitions between the interpolated data and the preexistent data. The extent of the problem depend

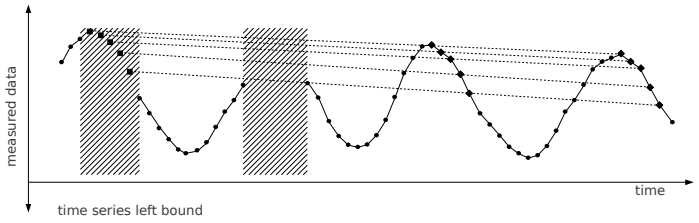


Figure 4.7: Conceptual representation of the optimization.

mostly from the stability of the climate and the relative difference between the meteorological data of consecutive seasonal cycles. Figure 4.8 gives a better conceptual representation of the problem. The interpolated points are represented as squares, while the points used for the interpolation are represented as rhombuses.

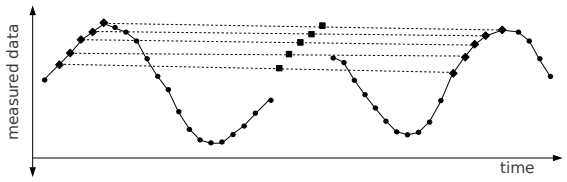


Figure 4.8: Conceptual representation of an issue with the Case 2 (single missing time series season) interpolation algorithm.

The proposed solution to this problem consists in fitting the interpolated data on the time series, such that the transition between the interpolated data and the preexistent data behave smoothly like it does one seasonal cycle away, in either direction, from the transition point.

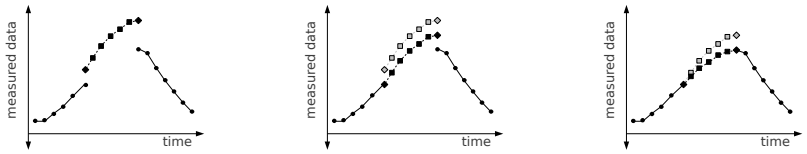
This can be done using a three step approach:

1. Extend the scope of the data segment that is going to be interpolated by two points, one on each side. After the interpolation, this results in a total of two pairs of overlapping points. Note that these two additional points get interpolated in the same way as all the other ones in the data segment. This first step is conceptually represented in Figure 4.9(a). The interpolated points are represented as squares. The points that have been added to the data segment and overlap with the two preexisting points are represented as rhombuses.

2. Adjust the value of all the interpolated points by the same extent such that, either on one or the other side, the values of the overlapping points coincide. This second step is conceptually represented in Figure 4.9(b). The interpolated points are represented as squares. The points that have been added to the data segment and overlap with the two preexisting points are represented as rhombuses. The value of the left one coincides with the preexistent point. The light drawn points show the situation before the first adjustment.

3. Adjust the value of every interpolated point by the weighted difference between the two overlapping points whose values do not coincide. The weight is the distance between the point being adjusted and the two overlapping points whose values coincide, over the

whole length of the data segment. This third step is conceptually represented in Figure 4.9(c). The interpolated points are represented as squares. The points that have been added to the data segment and overlap with the two preexisting points are represented as rhombuses. Now the value of both ones coincide with the preexistent points. The light drawn points show the situation before the second adjustment.



(a) (b) (c)

Figure 4.9: Conceptual representation of the adjustment steps.

This solution, since it requires known data on both sides of the empty data segment, is not directly applicable where the empty data segment lies directly at the beginning or at the end of the time series. In these cases it is only possible to extend the scope of the data segment that is going to be interpolated by one point. The value of the interpolated points can then be adjusted as described before, making coincide the two overlapping points. At this point, the transition between the interpolated data and the preexistent data already behave smoothly, without the need of applying the second adjustment described above, although its application would not be possible. Figure 4.10 conceptually shows the adjustment in this situation. The interpolated points are represented as squares. The point that has been added to the data segment and overlap with the preexisting point is represented as a rhombus. The light drawn points show the situation before the adjustment.

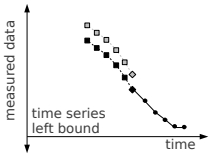


Figure 4.10: Conceptual representation of the adjustment in special cases.

4.3.3 Case 3: Multiple Missing Time Series Seasons

Although the MicroMet Case 3 (multiple missing time series seasons) interpolation algorithm and MSARIMA itself do substantially not show any deficiencies, there is still some room for improvements. As described in Chapter 3.2.2 “Case 3: Multiple Missing Time Series Seasons” MSARIMA uses a data segment of the same length of the empty data segment in order to do the forecasting/backcasting. In many cases it can be observed that MSARIMA does a more reliable forecasting/backcasting if the used data segment is increased by a factor of around five. This is, after all, also a requirement for the improvements that are going to be proposed in the following paragraphs.

In many cases, it can also be observed, that the average amplitude of the MicroMet Case 3 (multiple missing time series seasons) interpolated data segments tend to slightly increase

or decrease when compared to the preceding and succeeding data. Also the values at the local extrema seem not to coincide with the preceding data, as conceptually shown in Figure 4.11. The interpolated segment is represented by the hatched area. The dotted lines show the average extrema for the interpolated segment and for the preceding data.

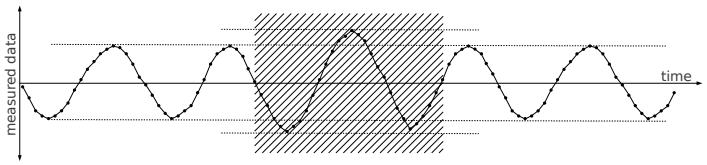


Figure 4.11: Conceptual representation of possible MSARIMA behaviors.

The suggested solution is to adjust the interpolated segment, acting on the local extrema, in order to make it fit better with the the data around it. The first step consists in determining the extrema, e.g. the local maximums, in the interpolated segment and in the two segments used by MSARIMA. The optimal value for each maximum inside the interpolated segment would be where the value coincides with the linear interpolation between the average of the local maximums preceding the interpolated segment and the average of the local maximums succeeding it. While calculating the average on both sides, it is usually more convenient, for achieving better results, to use only the two or three maximums for each side, which are close to the interpolated segment. Following, the optimal adjustment for each maximum inside the interpolated segment is the difference between actual value and optimal value. See Figure 4.12 for clarifications. The interpolated segment is represented by the hatched area. a is the optimal adjustment.

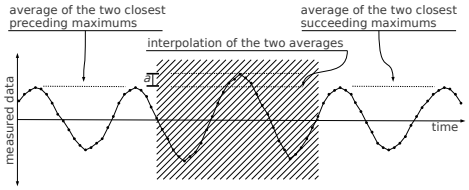


Figure 4.12: Conceptual representation of the adjustment.

The application of the optimal adjustment on every maximum in the interpolated segment would generate distortions in the time series, especially in the proximity of the bounds of the interpolated segment. The application of the optimal adjustment should be limited to the maximums that distance more than about one third of a seasonal cycle from both bounds. On the rest of the maximums inside the interpolated segment the weighted optimal adjustment should be applied. As weight, the distance from the nearest bound to the maximum point, which is going to be adjusted, over one third of a seasonal cycle should be used. In this way, the value of the maximums that reside near to the bounds of the interpolated segment are adjusted by a much smaller amount than the value of the maximums that are more far away. See Figure 4.13 for clarifications. The interpolated segment is represented by the hatched areas. The maximums inside the simple hatched area are going to be adjusted using the optimal adjustment. After the adjustment, its values will coincide with the interpolation of the maximum averages on both sides. The maximums inside the cross hatched area are adjusted using the weighted optimal adjustment. After the adjustment, its values will not

coincide with the interpolation of the maximum averages on both sides, but will be closer to it as before. a is one third of a seasonal cycle. b is the distance from the nearest bound to the maximum point.

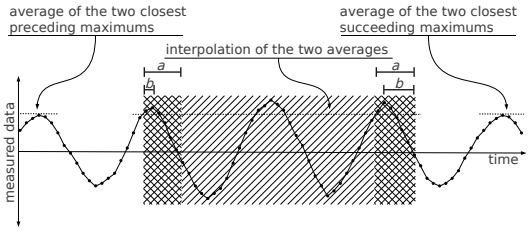


Figure 4.13: Conceptual representation of the adjustments.

Having determined the adjustment for the maximums, also the points around them have to be adjusted. For each maximum, the points that distance no more of one fourth seasonal cycle from both sides have to be adjusted. The adjustment that has to be applied to a maximum has to be applied trough a weight also to the points inside the two segments of one fourth seasonal cycle length. As weight, by taking e.g. the segment preceding the maximum, the distance from the segment's bound, which most distances from the maximum, to the point, which is going to be adjusted, over one fourth seasonal cycle, has to be used. The weight is calculated in an analogous way for the segment succeeding the maximum point. A conceptual representation of the adjustment is shown in Figure 4.14. The two segments of one fourth seasonal cycle length are represented by the hatched areas. The light drawn points represent the situation after the adjustment. a is one fourth of a seasonal cycle. b is the distance from the segment's bound, which most distances from the maximum, to the first point, which is going to be adjusted. c is the adjustment of the value of the maximum point.

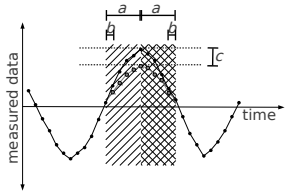


Figure 4.14: Conceptual representation of the adjustment of a maximum point.

The previously described optimization of the local maximums has to be analogously applied on the local minimums. This way the interpolated segment becomes more coherent with the time series data preceding and succeeding it, making it look less artificial and more realistic.

A major difficulty in applying the described optimization resides in the fact that meteorological data actually do not behave like a clear sine wave as shown in the conceptual representations until now, but presents many slight variations and multiple local extrema per seasonal cycle. This makes it difficult to distinguish between the various extrema and detect only those which are relevant for the previously described optimization. An example is shown in Figure 4.15.

A possibility of detecting the relevant extrema, is to do it by exclusion, iterating trough the time series and marking the points either as excluded or as extrema until all points in the time

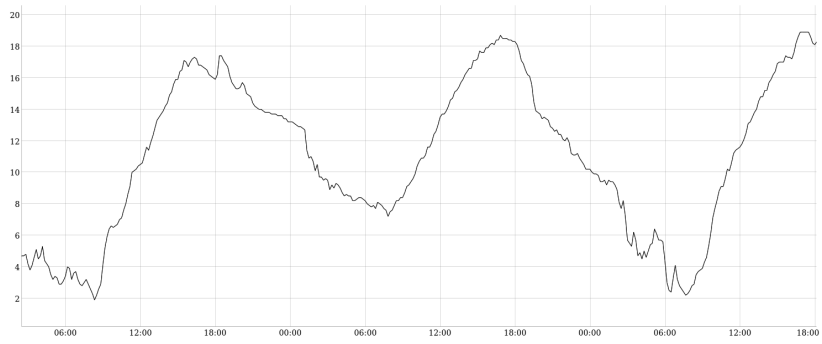


Figure 4.15: Example of real meteorological data showing air temperature.

series are marked in either way. This procedure has to be performed separately for detecting either maximums or minimums. In order to detect maximums, while continuously iterating trough the time series, the point with the lowest value which is not already marked is taken. In order to detect minimums, the unmarked point with the highest value is taken. If the point is the only one inside a predefined interval, it has to be considered an extreme and marked as such, in the other case it can be marked as excluded. The interval previously mentioned, in an optimal case, would range from one seasonal cycle before the point to one seasonal cycle after the point, but since the detected extrema usually not distance each other exactly one seasonal cycle, the interval should be smaller in order to avoid undetected extrema. The interval ranging from two thirds seasonal cycles before the point to two thirds seasonal cycles after the point has shown to work just reliably enough without compromising the quality of the result and still being able to detect all extrema in almost all the situations.

4.3.4 Efficacy Optimizations

Additionally to the already presented main optimizations to the MicroMet Preprocessor, there are some smaller optimizations that contribute to the flexibility and the ability of the model to exploit better the known data in the time series and fill time series with a higher frequency of data gaps.

In situations, where certain combinations of missing data segments occur, such that the interpolation of a specific missing data segment depend from data segments, which themselves contain missing data, and therefore from the ability of successfully interpolating other missing data segments, the MicroMet Preprocessor may not be able to interpolate all the missing data segments. Since, essentially, the interpolation of a specific missing data segment can depend on data that itself is going to be interpolated, the simple multiple application of the interpolation algorithms can solve this issue. A possible situation is conceptually represented in Figure 4.16. The hatched areas represent the empty data segments. Under the assumption, the limit of iterations for finding useful data segments for the Case 2 (single missing time series season) interpolation is set to two, a unique application of the Case 2 (single missing time series season) algorithm would not be able to fill all the missing data segments. It would first interpolate the segment in the middle, then the segment on the right and leaving the left segment unfilled. Only in a second application the algorithm would fill the left segment and

so outputting a continuous time series.

After the first consecutive application of the three interpolation algorithms, they should consecutively be applied again and again until no change to the time series is done anymore. It is important that the three interpolation algorithms are applied consecutively, since a multiple application of the first algorithm followed by a multiple application of the second and a multiple application of the third, would still leave missing data segments, which otherwise could be filled. Although the described procedure is expected to terminate, it is still reasonable to set an upper limit to the times the algorithms can be applied.

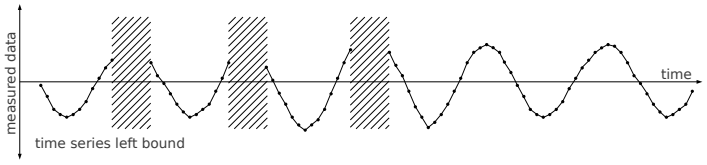


Figure 4.16: A possible situation where a unique application of the Case 2 (single missing time series season) algorithm is not able to fill all the missing data segments.

5 Implementation

The optimized MicroMet Preprocessor has been implemented entirely using Java. Since an optimized MicroMet Preprocessor is not reasonable as a standalone application, it has been thought and implemented as a black box application surrounding the core functionalities, which can easily be integrated or used as an external library in other applications handling with the problem of missing data segments in time series. These core functionalities have been programmed entirely in Java without auxiliary libraries or dependencies. So also the MSARIMA algorithm has been entirely coded without the usage of external libraries, this allows better customization and can make the code more flexible for integration in other applications.

The application, as it can be seen in Figure 5.1, after doing a series of checks regarding the validity of the time series, consecutively applies the filtering and the interpolation algorithms.

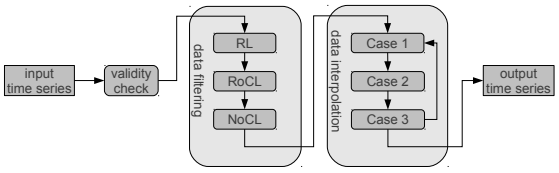


Figure 5.1: Data flow diagram of the application.

For simplicity, the application has been coded as a command line program. All inputs, as well as all outputs are read from and written to files, which are passed to the application using parameters. Beside the application's main input, the time series, which is read from file, it expects a configuration file specifying the parameters for the interpolation algorithms, the limits for the filtering algorithms, as well as some additional options.

The processed inputs are again outputted to file. In order to facilitate the analysis of the outputted time series, avoiding the inconvenience of a graphical user interface, the application generates an HTML file which visualizes the output using an open source JavaScript library called Dygraphs. Dygraphs produces vertically and horizontally zoom-able charts allowing to interact with the visualization.

Further, in order to allow evaluation, the application expects as input also a reference time series, which is supposed to be free of missing data gaps, as a comparison to the interpolated time series. The outputted HTML file can be created with both time series overlapping each other, the interpolated and the reference one, showing either the full range of the time series, or only the interpolated segments.

Depending on the configuration, the application also outputs the average difference between the interpolated points and the reference time series for each segment.

Figure 5.2 and 5.3 show screen shots of the rendered HTML file using Dygraphs. The interpolated segment shown on Figure 5.2 has been interpolated using the Case 2 (single missing time series season) algorithm. The interpolated segment shown on Figure 5.3 has been interpolated using the Case 3 (multiple missing time series seasons) algorithm. The dark line represents the reference time series, while the light line represents the interpolated data section.

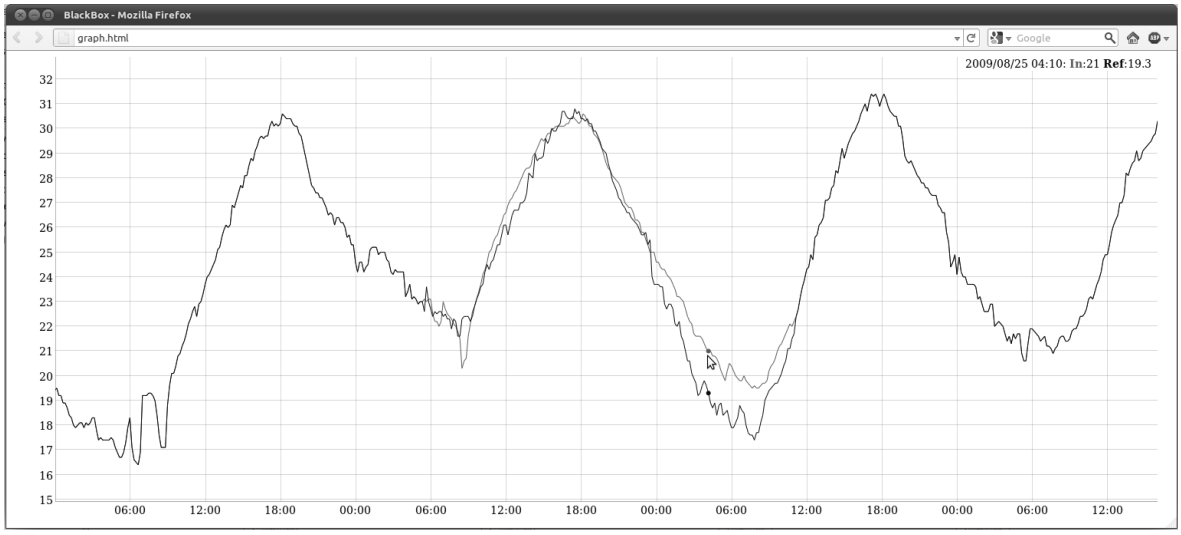


Figure 5.2: Screen shot of the rendered HTML using Dygraphs.

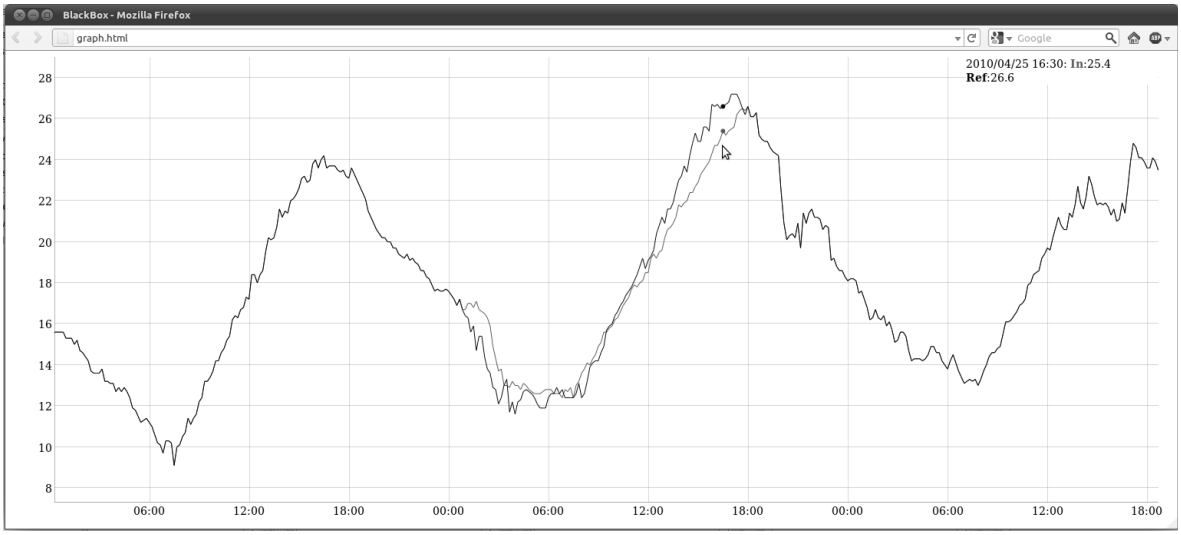


Figure 5.3: Screen shot of the rendered HTML using Dygraphs.

6 Empirical Evaluation

6.1 Methodology and Data Sets

As part of the research, this chapter discusses the outcomes of the MicroMet Preprocessor optimizations. The scope is to demonstrate, in an empirical way, that the optimizations made to the MicroMet Preprocessor actually produce better results and more realistic interpolated data that are closer to reality than the original MicroMet Preprocessor. The evaluation has been organized in three experiments.

For this purpose, real meteorological data has been used. The time series used for the experiment 1 and 2 have been measured in several places around Southtyrol and describes the air temperature in degree Celsius between January 2008 and November 2010 in a 10 or 30 minutes interval using a precision of one fractional digit. The characteristics of those time series are shown in Table 6.1.

Time series # [measurement point]	**# of missing data segments**	**# of points in each missing data segment**	**Minimal # of points between missing data segments**	**Interval in time between each point**	**Length in time of each missing data segment [interpolation algorithm]**
1 [Bozen]	501	6	300	10 min.	1 h. [Case 2]
2 [Bozen]	457	36	300	10 min.	6 h. [Case 2]
3 [Bozen]	414	72	300	10 min.	12 h. [Case 2]
4 [Bozen]	375	108	300	10 min.	18 h. [Case 2]
5 [Bozen]	350	138	300	10 min.	23 h. [Case 2]
6 [Bozen]	154	145	870	10 min.	24 h. [Case 3]
7 [Bozen]	80	289	1734	10 min.	48 h. [Case 3]
8 [Bozen]	54	433	2598	10 min.	72 h. [Case 3]
9 [Bozen]	42	577	3462	10 min.	96 h. [Case 3]
10 [Branzoll]	350	72	300	10 min.	12 h. [Case 2]
11 [Branzoll]	297	138	300	10 min.	23 h. [Case 2]
12 [Branzoll]	63	289	1734	10 min.	48 h. [Case 3]
13 [Branzoll]	31	577	3462	10 min.	96 h. [Case 3]
14 [Ladurns]	408	72	300	10 min.	12 h. [Case 2]
15 [Ladurns]	346	138	300	10 min.	23 h. [Case 2]
16 [Ladurns]	74	289	1734	10 min.	48 h. [Case 3]
17 [Ladurns]	37	577	3462	10 min.	96 h. [Case 3]
18 [Radein]	406	24	100	30 min.	12 h. [Case 2]
19 [Radein]	344	46	100	30 min.	23 h. [Case 2]
20 [Radein]	74	96	578	30 min.	48 h. [Case 3]
21 [Radein]	36	192	1154	30 min.	96 h. [Case 3]

Table 6.1: Characteristics of the time series used for the evaluation.

The implementation of the MicroMet Preprocessor has been run over these time series

with a series of generated missing data segments, both with and without the optimizations proposed in this thesis work. The missing data segments have been created in such a way, that the original MicroMet Preprocessor, which is not capable of handling a number of special cases, is still able of interpolating all of them. Moreover, only Case 2 (single missing time series season) and Case 3 (multiple missing time series seasons) algorithms have been tested, since the Case 1 (single missing time series point) optimizations actually do not improve data quality but only makes it possible to handle a number of special cases. The optimized preprocessor has been run using the following configuration for the Case 3 (multiple missing time series seasons) optimization:

- For the MSARIMA input, a factor of five times the missing data segment length has been used.
- The number of extrema used for the averages of both sides of the missing data segment have been limited to two.
- The size of the interval, where the weighted optimal adjustment has to be applied, has been set to eight hours.
- The size of the intervals on both sides of each extrema, where the interpolated data has to be adjusted, has been set to six hours.
- The size of the interval used for exclusion while detecting the extrema has been set to sixteen hours on both sides.

For details see Chapter 4.3.3 "Case 3: Multiple Missing Time Series Seasons". This configuration has been chosen such that it fits best for the time series measured in Bozen.

The interpolated segments of the outcomes of both runs have been compared with the original integral time series. For the experiment 1, the average of the difference between each interpolated and original point inside every segment has been calculated. Finally the arithmetic mean of all the segment's averages has been computed. The unit of the outcome is $^{\circ}C$. Further, the maximum difference for every time series has been included in the evaluation. For experiment 2, the frequency distribution per measurement point of the differences has been used.

The optimizations made to the MicroMet Preprocessor, not only aim to improve the quality of the interpolated data, but aim also to the ability of working correctly on time series presenting a high frequency of missing data segments. In the experiment 3, in order to test these kind of optimizations, both the optimized and the original MicroMet Preprocessor have been used to interpolate trough 100 time series with randomly generated missing data segments. These time series had, between 19,49% and 23,14%, and on average 21,48% of missing points. These points were distributed over an average of 477 missing data segments. The size of the missing data segments have been limited to 24 hours and interpolated using only the Case 1 (single missing time series point) and Case 2 (single missing time series season) algorithms, since the Case 3 (multiple missing time series seasons) algorithm actually does not make it possible to handle special cases, but improves data quality. The number of preprocessing cycles has not been limited. The limit of iterations for finding useful data segments for the Case 2 (single missing time series season) interpolation has been set to two (See Chapter 4.3.4 and 4.3.2).

6.2 Results

The previously described evaluation methods for testing the interpolated data quality have been applied on a number of different time series with different sizes of missing data segments. The actual results are shown in Figure 6.1 and 6.2.

Experiment 1. Figure 6.1 shows the average and the maximum difference between each interpolated and original point, both for the optimized and the original MicroMet Preprocessor, for each time series. It is possible to infer that the Case 2 (single missing time series season) optimizations achieve a relatively big improvement on missing data segments shorter than twelve hours, while on longer segments, up to 24 hours, the improvement is smaller but still significant. Instead, the Case 3 (multiple missing time series seasons) optimizations tend to achieve better results while increasing the segments length.

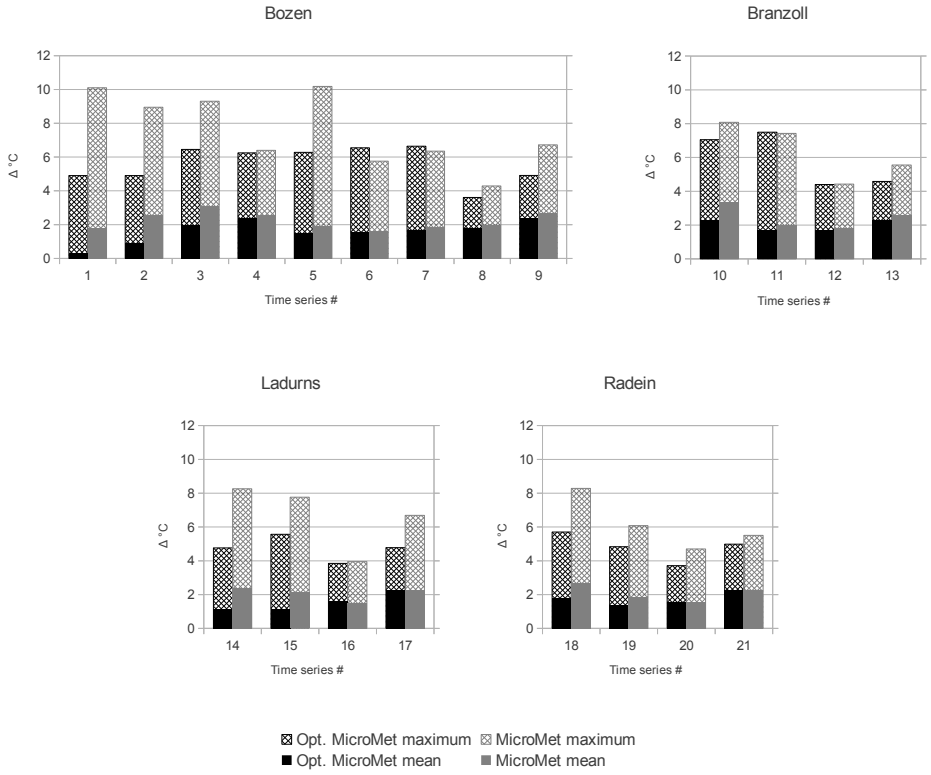


Figure 6.1: Average differences.

Experiment 2. Figure 6.2, shows the frequency distribution of all the differences between the interpolated and original points for all the time series for each measurement point. It is possible to see that the frequency of small differences, between 0.0 $°C$ and 2.0 $°C$ have generally increased, in comparison to the original MicroMet Preprocessor, while the frequency of all other bigger differences have decreased.

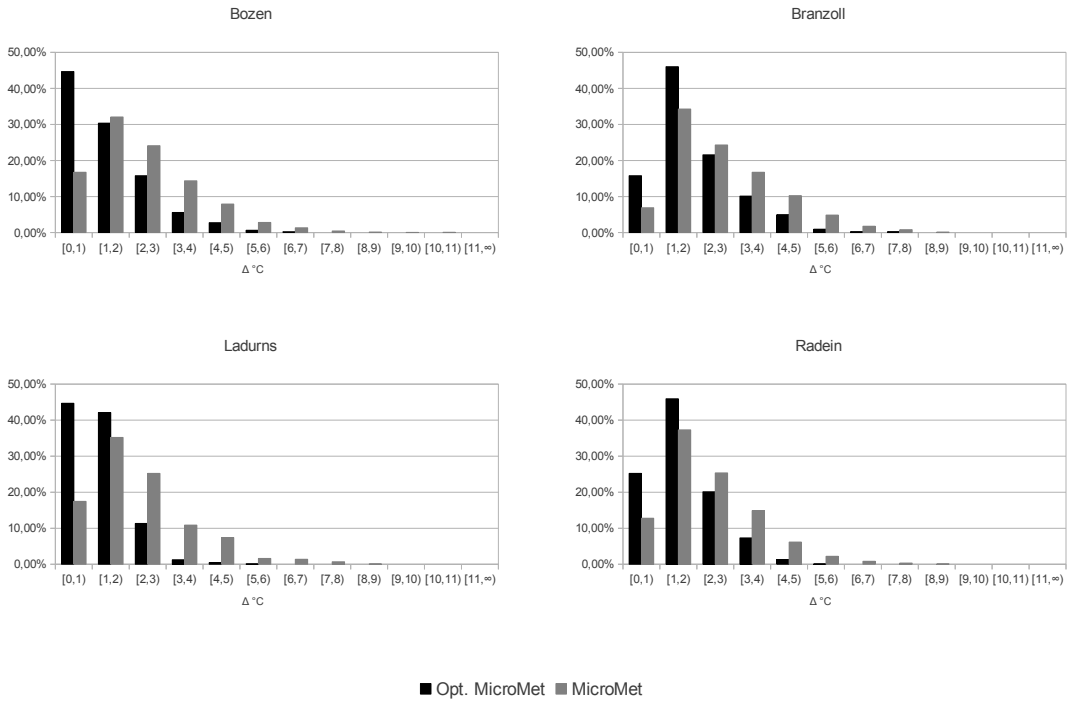


Figure 6.2: Frequency distribution of the differences.

The results of the evaluation show that the optimized MicroMet Preprocessor actually produces on average better results than the original MicroMet Preprocessor in the great majority of the tested situations. It also can be noted that the results of the tests done on time series measured in Bozen are slightly better than the results of the other tests. This is due to the fact that, as previously mentioned, the Case 3 (multiple missing time series seasons) configuration has been optimized for them.

Experiment 3. As previously described, in order to test the special case handling, both the optimized and the original MicroMet Preprocessor have been used to interpolate trough multiple time series with randomly generated missing data segments. The optimized MicroMet Preprocessor has been able to interpolate over 100% of the missing data, returning a continuous time series. The original MicroMet Preprocessor instead has been able to cover on average only 50,87% of the missing points by filling 57,38% of the missing data segments.

It should also be considered that the high frequency of missing data segments did not influence the quality of the data produced by the original MicroMet Preprocessor and the optimized MicroMet Preprocessor in the cases where it did not make use of its special case handling capabilities. Instead, the remaining data generated in special cases by the optimized MicroMet Preprocessor, about 49,13%, has shown to be not as good. A direct comparison of its quality with the previously reported results can not be done, since the missing data segments were generated of random length. Further, it should be noticed that time series with such a high frequency of missing data segments are not realistic an are not expected to be found in practice.

7 Conclusion and Future Work

The main goal of this thesis project was to implement the MicroMet Preprocessor meteorological model in an application capable of interpolating missing data segments in seasonal time series, further, to optimize its algorithms to produce more accurate data that reflects better the reality and to extend them making the application more flexible and capable of handling a larger number of situations. This goal has been successfully attained and the application has been pulled trough a series of tests to prove the actual improvement from the original MicroMet Preprocessor. The tests have shown that the optimized MicroMet Preprocessor achieves better results than the original model in the majority of the cases and that also its capability of interpolating time series with a high density of missing data segments has benefited.

Although the previously presented results already show that the interpolated data quality benefits from the presented optimizations, there is still margin for improvement.

The Case 3 (multiple missing time series seasons) optimization actually do not differ from the original algorithm when it comes to special cases. Both need a relatively big segment on both sides of the missing data segment in order to run MSARIMA, and there may be situations where these segments are either not continuous or not available. Such situations occur when a missing data segment, on which Case 3 (multiple missing time series seasons) should be applied, is found near to the time series bounds or when a data segment between two missing data segments, both for Case 3 (multiple missing time series seasons) application, is too small to run MSARIMA. It is also to consider that the Case 3 (multiple missing time series seasons) optimization need a bigger MSARIMA input data segment than the original MicroMet algorithm. A possible improvement could be to interpolate the missing data segment by doing either the forecasting or the backcasting, instead of doing both, and finally apply a similar adjustment as in the Case 2 (single missing time series season) optimization. The optimal adjustment should then be calculated by extrapolation instead of interpolation between the extrema. If in some situations this still does not make it possible to interpolate, the algorithm could fall back and switch off the interpolated data adjustment, decreasing also the size of the needed data segment for MSARIMA. Of course, also a combination of the two possibilities would work.

MSARIMA and also the general ARIMA model are well known statistical algorithms, which actually aim to analyze, describe and represent existing time series. The model can be fitted to a specific time series by adjusting various variable parameters. Inverting this procedure, under some assumptions, allows to use the previously analyzed data to do forecasting. Both the optimized and the original MicroMet Case 3 (multiple missing time series seasons) algorithm do not adapt these parameters before forecasting, instead, they both use fixed values that have shown to work in an acceptable way on these kind of time series. The estimation of these parameters would lead to an additional improvement, although, depending on the time

series, it probably could be not very significant.

As for MSARIMA, the Case 3 (multiple missing time series seasons) optimization has several parameters that could be adjusted. Finding a way, for choosing the right values, such that they fit best to the time series would additionally improve the interpolated data quality.

A last interesting topic for further research could be the application of the optimized Case 3 (multiple missing time series seasons) algorithm to missing data segments, whose length is less than a seasonal cycle. As it is possible to see from the previously presented results, the Case 3 (multiple missing time series seasons) algorithm is more accurate on 24 hours gaps than the Case 2 (single missing time series season) algorithm on 23 hours gaps. This fact suggests the possibility for further improvement.

References

[1] G. E. P. Box and G. M. Jenkins. *Time Series Analysis - Forecasting and Control*. Holden Day, 1970.

[2] S. R. Brubacher and G. T. Wilson. Interpolating time series with application to the estimation of holiday effects on electricity demand. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 25(2):107–116, 1976.

[3] R. Cheng and M. Pourahmadi. Prediction with incomplete past and interpolation of missing values. *Statistics & Probability Letters*, 33(4):341–346, 1997.

[4] JGrass. A hydro-geomorphologic plugin for uDig. Retrieved from: `http://code.google.com/p/jgrass/`.

[5] JGrasstools. A library containing all geoprocessing power behind JGrass. Retrieved from: `http://code.google.com/p/jgrasstools/`.

[6] Glen E. Liston and Kelly Elder. A distributed snow-evolution modeling system (SnowModel). *Journal of Hydrometeorology*, 7:1259–1276, 2006.

[7] Glen E. Liston and Kelly Elder. A meteorological distribution system for high-resolution terrestrial modeling (MicroMet). *Journal of Hydrometeorology*, 7:217–234, 2006.

[8] MatLab. A programming environment for algorithm development, data analysis, and numerical computation. Retrieved from: `http://www.mathworks.com/products/matlab/`.

[9] MeteoIO. A library making data access easy and safe for numerical simulations in environmental sciences requiring general meteorological data. Retrieved from: `http://slfsmm.indefero.net/p/meteoio/`.

[10] GNU Octave. A high-level interpreted language, primarily intended for numerical computations. Retrieved from: `http://www.gnu.org/software/octave/`.

[11] M. Pourahmadi. Estimation and interpolation of missing values of a stationary time series. *Journal of Time Series Analysis*, 10(2):149–169, 1989.

[12] GNU R. A free software environment for statistical computing and graphics. Retrieved from: `http://www.r-project.org/`.

[13] SEST. Approximation of missing data in time series. Retrieved from: `http://www.ifi.uzh.ch/dbtg/research/amv.html`.

[14] Mehmet Tektas. Weather forecasting using ANFIS and ARIMA models. a case study for istanbul. *Environmental Research, Engineering and Management*, 51(1):5–10, 2010.

[15] Augustin Maravall Victor Gomez and Daniel Peña. *Computing Missing Values in Time Series*. Working Paper. Universidad Carlos III de Madrid, Departamento de Estadistica y Econometria, 1993.

[16] Swarna Weerasinghe. A missing values imputation method for time series data: an efficient method to investigate the health effects of sulphur dioxide levels. *Environmetrics*, 21:162–172, 2010.