Restructuring and harmonizing the code used to calculate the Definitive Polar Cap Index

Date: 31 October 2019

Prepared by Jonas Bregnhøj Nielsen, Data analyst, DTU Space

Checked by Anna Naemi Willer, Academic staff, DTU Space

Table of Contents

1 Introduction 3
2 Applicable and Reference Documentation 3
2.1 Applicable Documents 3
2.2 Reference Documents 3
3 Software Improvement Summary 4
4 Detailed Description of Software Improvement 5
4.1 Software Improvements 5
4.1.1 MySQL Database 5
4.1.2 Edge Effects around New Year 5
4.1.3 Quiet Daily Curve 7
5 Validation 9
5.1 Impact on the Sector Structure 9
5.2 Impact on the Quiet Daily Curve 11
5.3 Impact on the definitive PC index 12
6 Appendix A 14
6.1 Definitive_PC_Index.m 14
6.2 append_db.m 18
6.3 q_day.m 18

Table of Figures

Figure 4-1 Edge effects in the Sector Structure and Quiet Daily Curve around New Year 2000 6
Figure 4-2 Edge effects in the PCN-index around New Year 2000 6
Figure 4-3 Quiet Days using different smoothing methods 7
Figure 5-1 Difference in Sector Structure caused by changes in new code for 2000 and 2009 9
Figure 5-2 Sector Structure for 2000 and 2009 using original and improved code 10
Figure 5-3 Zoom in on Sector Structure for 2000 and 2009 using original and improved code 10
Figure 5-4 Difference in Quiet Daily Curve caused by changes in improved code for 2000 and 2009 11
Figure 5-5 Quiet Daily Curve x-component for July 2000 using original and improved code 12
Figure 5-6 Difference in definitive PC index caused by changes in improved code for 2000 and 2009 13
Figure 5-7 Changes in definitive PC index from 1975 until 2014 13

1 Introduction

In this document we present the restructuring and harmonizing implemented at DTU to the IAGA endorsed definitive Polar Cap (PC) index software [AD-2]. The updates address an overall optimization of the code as well as some minor corrections. By optimizing the code the time for reprocessing the entire definitive PCN (North) dataset is reduced from several days to less than one day. Moreover the number of scripts and sub-functions listed in [AD-2] has been reduced drastically, giving better code overview and understanding of the PCN calculation structure.

2 Applicable and Reference Documentation

2.1 Applicable Documents

The following documents are applicable to the definitions within this document.

[AD-1] IAGA_documentation, Supplying relevant supporting material for endorsement of Polar Cap index by the International Association of Geomagnetism and Aeronomy, 2013-02-25

[AD-2] Apendix A, Part of: Supplying relevant supporting material for endorsement of Polar Cap index by the International Association of Geomagnetism and Aeronomy, 2013-02-15

2.2 Reference Documents

The following documents contain supporting and background information to be taken into account when reading this document

[RD-1] Janzhura, A. and Troshichev, O., 2008. Determination of the running quiet daily geomagnetic variation. J. Atmos. Solar-Terr. Phys., 70, 962-972

[RD-2] Janzhura, A. and Troshichev, O., 2011. Identification of the IMF sector structure in near-real time by ground magnetic data. Ann. Geophys. 29, 1491-1500

3 Software Improvement Summary

The improvements addresses an overall optimization as well as some minor corrections of the code. The optimization primarily consists of a parallelization of many of the routines [AD-2], as well as a reduced interaction between Matlab and the MySQL database which is very time consuming. Most of the code is now contained in the `Definitive_PC_Index.m` script, only the new `append_db.m` and the updated `q_day.m` scripts are needed besides the main script (see Section 6, Appendix A). No changes have been made to the files `db_config.pc` and `coeff.mat`. This optimization has reduced the time for reprocessing the entire definitive PCN dataset from several days to less than a day. Besides the optimization, some corrections have been made to the code, primarily correcting for loosing of precision and intrinsic contradictions in the original code (see Section 4 below).

4 Detailed Description of Software Improvement

This section contains detailed descriptions of the changes made to the definitive PC index code [AD-2]. The number of interactions between Matlab and MySQL is reduced to make the code more time efficient. The number of significant digits used during the PC index determination is revised. To avoid edge effects, which could give rise to non-physical features around the beginning and the end of a year, a data overlap of one month is added to the yearly definitive PC calculations. Lastly the Quiet Daly Curve (QDC) part of the code has been updated to correct some inconsistencies between the algorithm description and its software implementation.

4.1 Software Improvements

4.1.1 MySQL Database

Interaction between Matlab and MySQL was one of the most time consuming part of the processing. In the original code all variables (*t*, *x*, *y*, *ss_x*, *ss_y*, *qdc_x*, *qdc_y* and *pc*) were saved into the database immediately after they are calculated, and retrieved again whenever used for a new calculation. This resulted in numerus interactions between Matlab and MySQL, moving variables back and forth, which is very inefficient and time consuming. In the improved code all variables (*t*, *x*, *y*, *ss_x*, *ss_y*, *qdc_x*, *qdc_y* and *pc*) for one year are preserved in a Matlab data structure, and only written to the MySQL database after calculation of a complete year of definitive PC indices. Note that the magnetic *Z*-component is no longer saved to the database as it is not used in the calculation of the definitive PC index.

4.1.1.1 Significant Digits

The type setting of the entries in the database for the variables two horizontal magnetic components stored in the variables *x* and *y* have been changed from *mediumint(9)* to *float*. As the PC index is given with a two decimal precision it was bad practice to round off the input data to integers, especially since the definitive IAGA2002 data are given with a precision of two decimals (0.01 nT). Moreover the original code saved the variables *ss_x*, *ss_y*, *qdc_x*, *qdc_y* to the database as rounded integers even though the type setting for these variables in the database are set as *float*. Afterwards the rounded variables are used for further calculations of the definitive PC index. In the updated code all variables (*x*, *y*, *ss_x*, *ss_y*, *qdc_x*, *qdc_y* and *pc*) are kept as *doubles* within Matlab throughout all calculations of the definitive PC index, and then saved to the database with a precision of two decimals.

4.1.1.2 MySQL Batch Insert

To further reduce the time spent on writing results to MySQL, entire sets of the variables *t*, *x*, *y*, *ss_x*, *ss_y*, *qdc_x*, *qdc_y* and *pc* are saved to the database within one INSERT statement:

```
INSERT INTO pcnthl (time, x, y, ss_x, ss_y, qdc_x, qdc_y, pc) VALUES (t, x, y, ss_x, ss_y, qdc_x, qdc_y, pc) ON DUPLICATE KEY UPDATE x = VALUES(x), y = VALUES(y), ss_x = VALUES(ss_x), ss_y = VALUES(ss_y), qdc_x = VALUES(qdc_x), qdc_y = VALUES(qdc_y), pc = VALUES(pc);
```

The INSERT statements are executed in batches of 1000 (the maximum number accepted by MySQL), dramatically reducing the time spent on writing results to the database.

4.1.2 Edge Effects around New Year

The smoothing schemes and extrapolations used in the code results in some edge effects around the beginning and end of the processing period. These effects are not taken into account in the original code, where the processing period was one year, thereby giving rise to some non-physical features around New Year (see Figure 4-1). To avoid these edge effects, a data overlap of one month is added to the yearly definitive PC calculations.

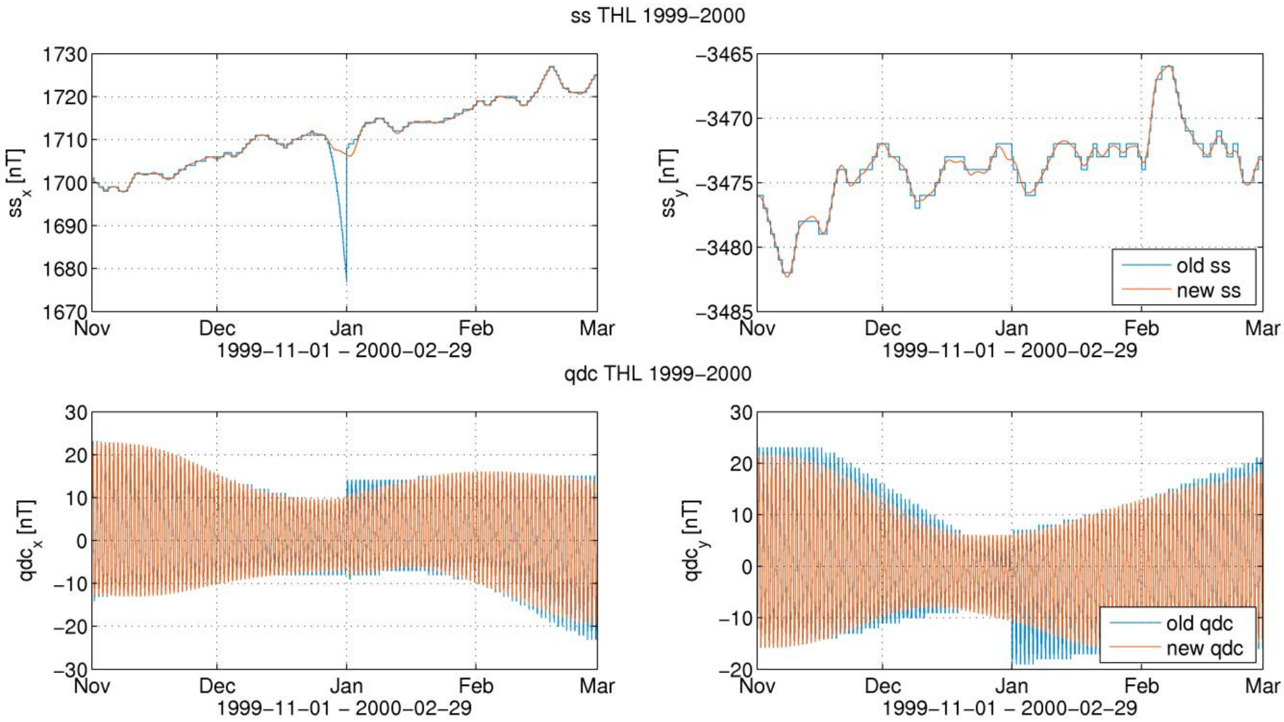

Figure 4-1 Edge effects in the Sector Structure and Quiet Daily Curve around New Year 2000

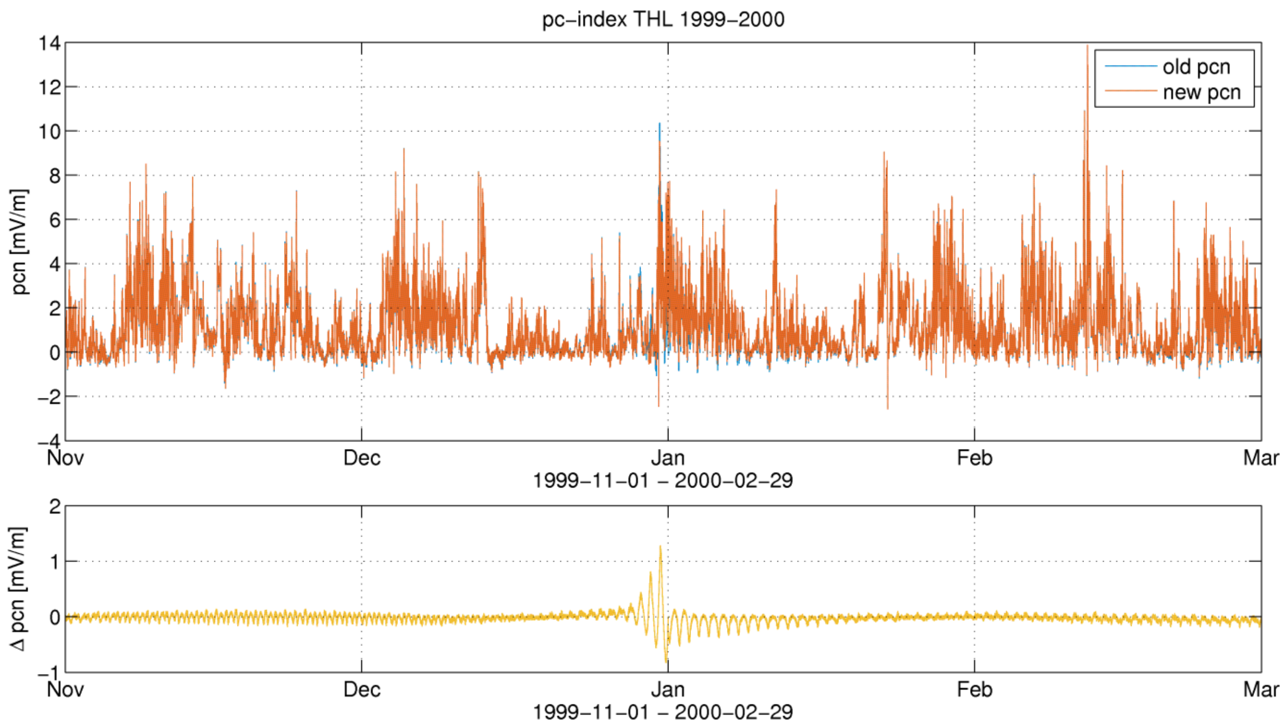

Figure 4-2 Edge effects in the PCN-index around New Year 2000

Figure 4-1 shows the Sector Structure (SS) and the Quiet Daily Curve (QDC) around New Year 2000 calculated with the old (blue line) and new (orange line) software. It is seen that the original software introduces a drift in the SS at the end of the year, resulting in a large jump at the beginning of the next year. It is also seen that the QDCs computed by the original software are non-continuous from one year to the next year. Both of these cases are primarily due to extrapolations. By adding one month of data as overlap before and after the year that is processed, these non-physical edge effects are removed. Figure 4-2 shows an impact on the definitive PC index around New Year 2000 of up to ±1 mV/m.

4.1.3 Quiet Daily Curve

Some corrections have been made to the QDC routine. In the computation of the actual Quiet Day, a *method* input has been added as argument to the smoothing subfunction when finding the 2 hour trend in the magnetic field. The actual QDC is now calculated for a 30 day period (instead of erroneously 31 days), as stated in [AD-2], and the Actual Day is assigned to the correct date.

4.1.3.1 Quiet Day correction

When determining the 2 hour trend in the magnetic field using a running smoothing value, the original code lacks the *method* input. When no *method* input is specified, the smoothing function uses the *moving* option as default if the dataset is uniformly spaced. However, if the dataset is not uniform and no *method* is specified *lowess* is used. As gaps do accrue in the definitive IAGA2002 data both methods are erroneously used in the previous code. Figure 4-3 shows actual Quiet Days determined using the different smoothing options in MATLAB. It is seen that determining the actual Quiet Day is quit sensitive to the chosen smoothing method when finding the 2 hour trend in the magnetic field. The different methods may even result in different identification of Actual Quiet days. To ensure that the QDCs and Actual Days are all calculated in the same way, the smoothing method is set to *moving* in the improved code.

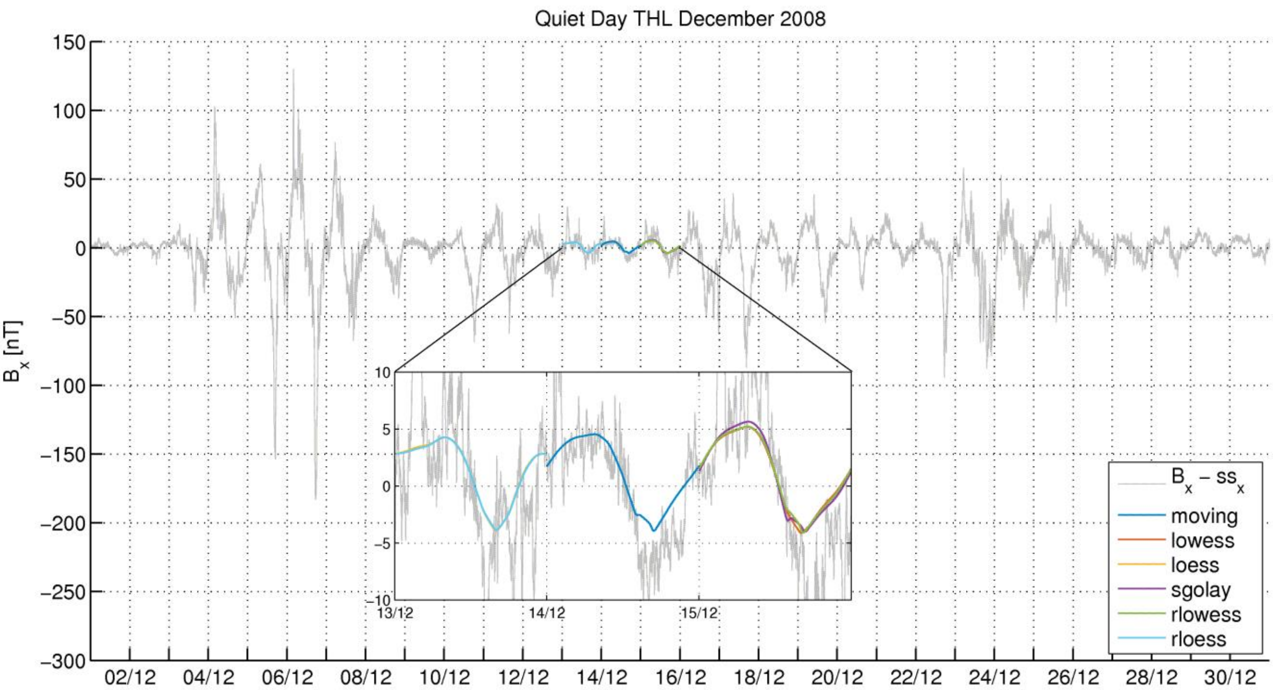

Figure 4-3 Quiet Days using different smoothing methods

Note that no specific description of the method to be used are mentioned in [AD-2], but by choosing the moving method the improved software agrees with the original one in the case of a uniform dataset.

4.1.3.2 Actual Day correction

In [AD-2] the calculation of the QDC is described: *"For the definitive PCN index a series of actual QDCs are calculated by shifting the 30-day period by 10 days at a time"*. However, in the original code, this 30 day period is found by adding 30 days to the start date of the period, resulting in a 31 day period instead:

```
% filed end date of 30-days period
ds = datenum(start_dt);
[y, m, d] = datevec(ds+30);
end_dt = [y,m,d,23,59,00];
```

In the improved code this period are determined by adding 30 days to the start date, then subtracting one minute giving a period of 30 days until the last minute of the last day:

```
for start_dt = datenum(year, 1, 1)-overlap:10:datenum(year, 12, 31)
    % Calculate QDC for 30-days dataset and save it into
    % "Actual Day" in data structure
    end_dt = start_dt+30-1/1440;

    ...

end%for start_dt
```

Further on, in the original code, the date of the Actual Day (the day for which the most quiet conditions were met) are found by adding the Actual Day within the 30 days (a number from 1 to 30) to the start date:

```
if ~isnan(ActDay_x)
     % Calculate date for the Actual Day;
     [y, m, d] = datevec(ds+ActDay_x);
     act_dt_x = [y, m, d];
      % Write QDC for Actual Day to DB
      dbset ( [act_dt_x,0,0,0],[act_dt_x,23,59,0],'qdc_x',qday_x);

end;
```

This results in a date just after the Actual Day, as this don't take the start date itself into account. In the improved code the date of the Actual Day is found by:

```
% Write QDC for "Atcual Day" to data structure
if ~isnan(ActDay) % If any
  j = data.t >= start_dt + ActDay - 1 & ...
      data.t <= start_dt + ActDay - 1/1440;
  data.(['qdc_' char(fn)])(j) = qday;
end%if ~isnan
```

which gives the date of the Actual Day.

5 Validation

In this section the impact of the software changes on the definitive PC index is presented and discussed. In general the changes have the largest impact on the calculations during periods of high activity, i.e. in years of higher solar activity and during summer. In the following, two different years are used as cases, namely year 2000 with high solar activity and year 2009 with low solar activity. All the changes in the improved code have been tested separately to show their individual impact on the definitive PC index calculations.

5.1 Impact on the Sector Structure

Two changes in the new code have an impact on the SS, namely the increased significant digits and the added one month data overlap to the computation. Figure 5-1 shows the discrepancies between the original and new SS. Figure 5-2 and Figure 5-3 shows the SS calculated with the original (blue line) and new (orange line) programs. Both Sector Structures generally follow the same pattern, with some lager differences during summer, especially in the solar maximum year 2000. Figure 5-1 reveals that these discrepancies primarily comes from the added one month of data overlap to the yearly processing (orange line). More precisely this is due to the smoothing scheme in the SS calculation:

```
% Smooth medians
w = smooth(day_med, 7, 'moving'); % 7 days
w = smooth(w, 7, 'rloess');
```

Here the *rloess* smoothing is sensitive to the size of the dataset, thus resulting in the difference seen in the two Sector Structures. Finally Figure 5-1 reveals that increasing the significant digits (yellow line) has only minor impact on the calculations.

Figure 5-1 Difference in Sector Structure caused by changes in new code for 2000 and 2009

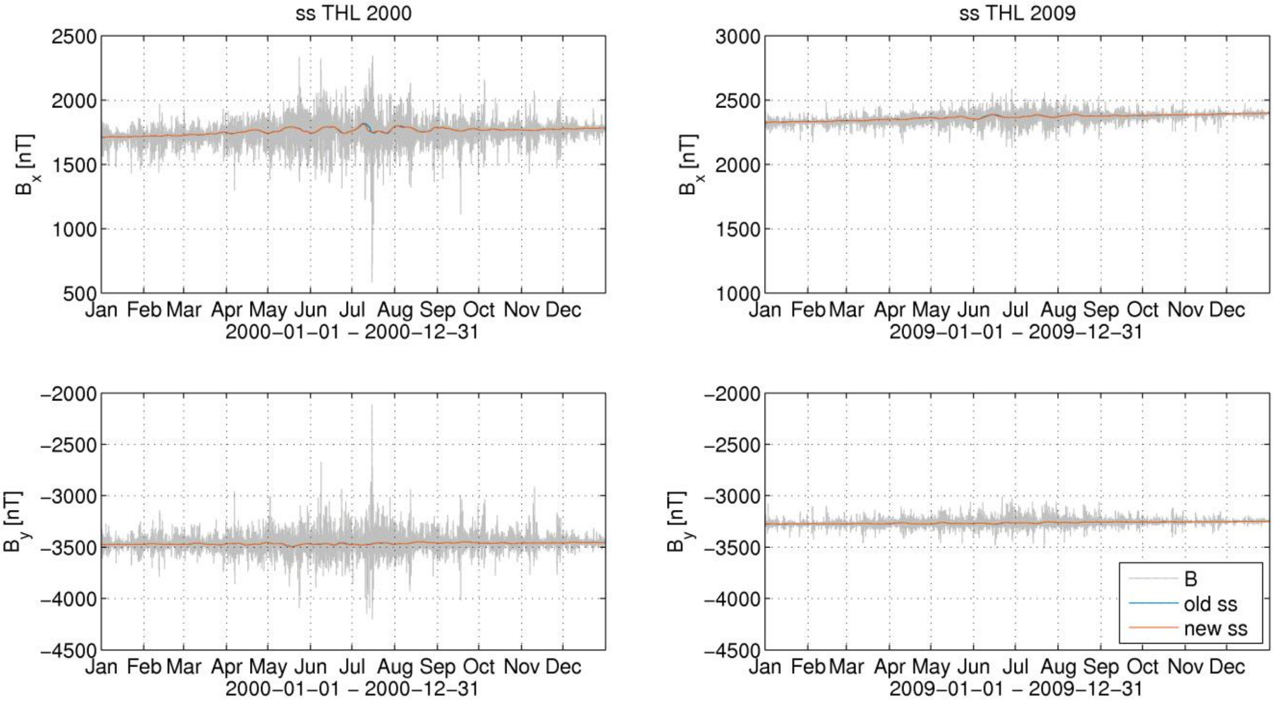

Figure 5-2 Sector Structure for 2000 and 2009 using original and improved code

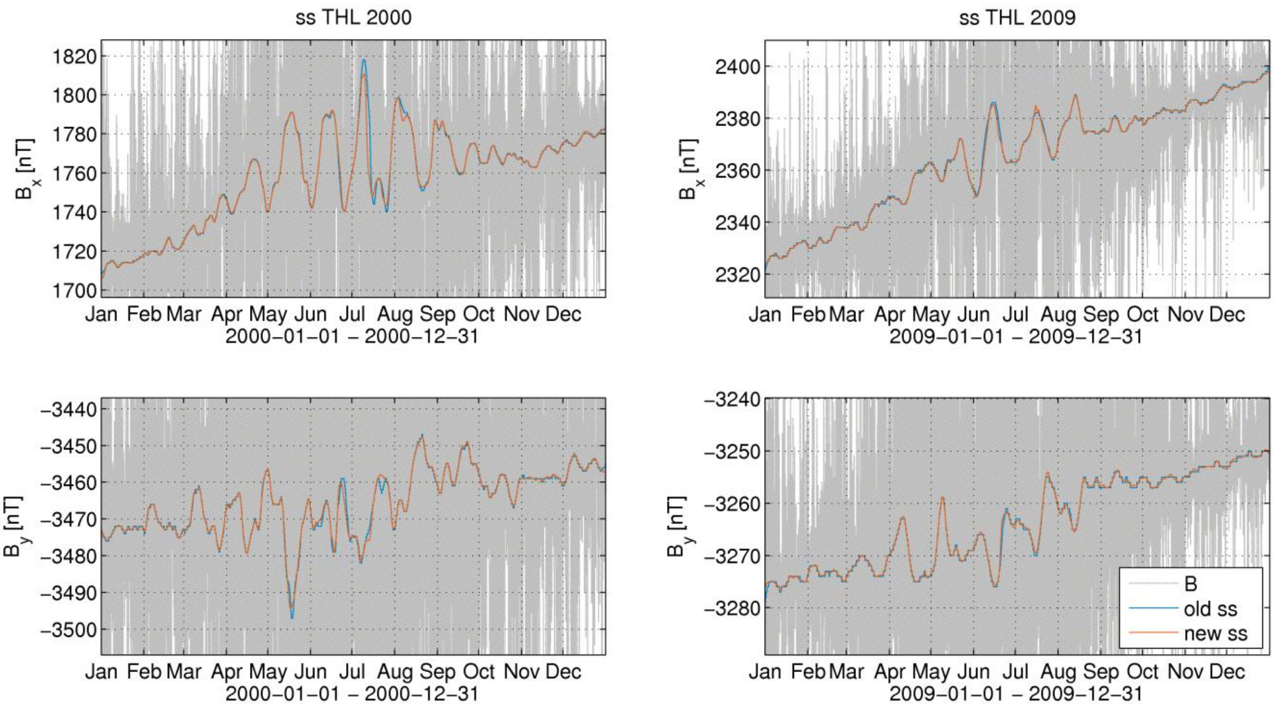

Figure 5-3 Zoom in on Sector Structure for 2000 and 2009 using original and improved code

5.2 Impact on the Quiet Daily Curve

All changes in the improved code have an impact on the QDC to some degree (See Figure 5-4). The most significant are the added *moving* setting to the smoothing method used when finding the 2 hour trend in the magnetic field during the QDC computation (orange line). The added one month data overlap (yellow line) also has some impact, but this is a direct consequence of the discrepancies in the SS, as the QDC are computed from the magnetic field after subtracting the SS. The discrepancies around New Year are due to the corrections of the non-physical edge effects by adding the data overlap. As shown in Section 4.1.3.1 the actual QDCs are quit sensitive to the smoothing method used when finding the 2 hour trend in the magnetic field. Different methods may even result in different Actual Quiet days. This correction in the new code causes some larger discrepancies during active periods, i.e. during summer months in years of high solar activity. Figure 5-5 shows the two different QDCs for the X-component during July 2000 (solar maximum year). In this case the new code finds a more “quiet” QDC.

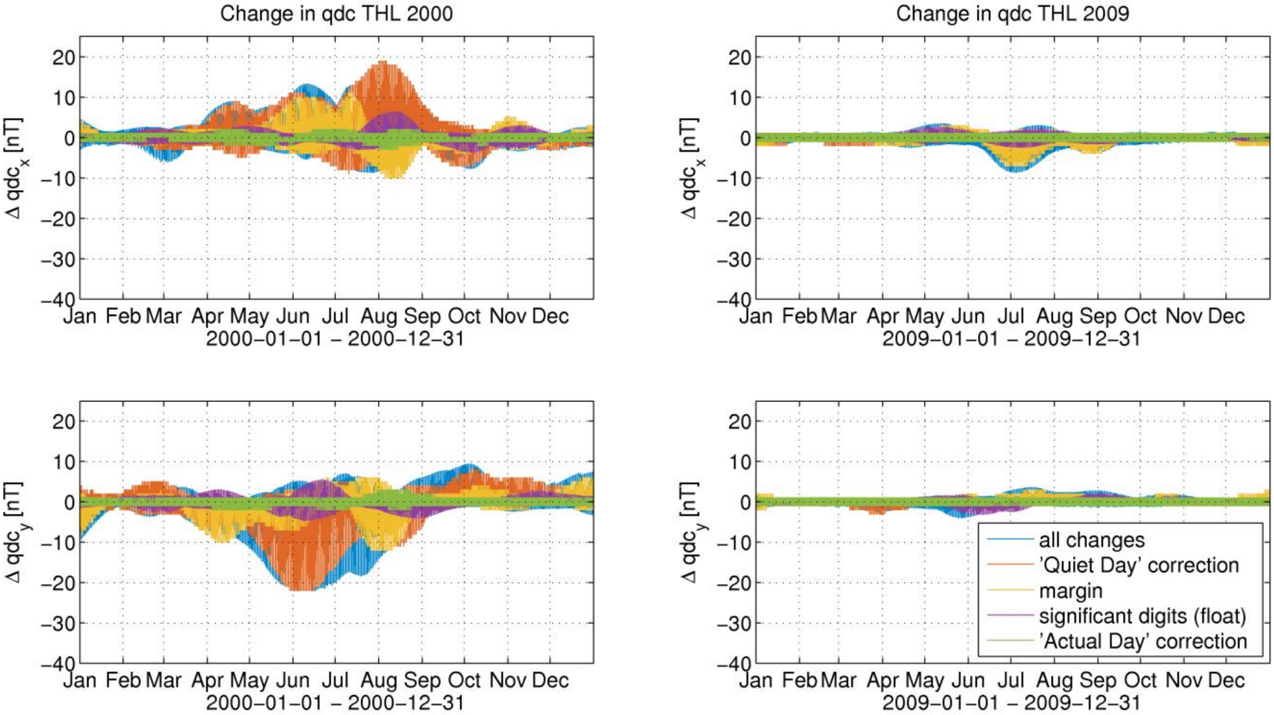

Figure 5-4 Difference in Quiet Daily Curve caused by changes in improved code for 2000 and 2009

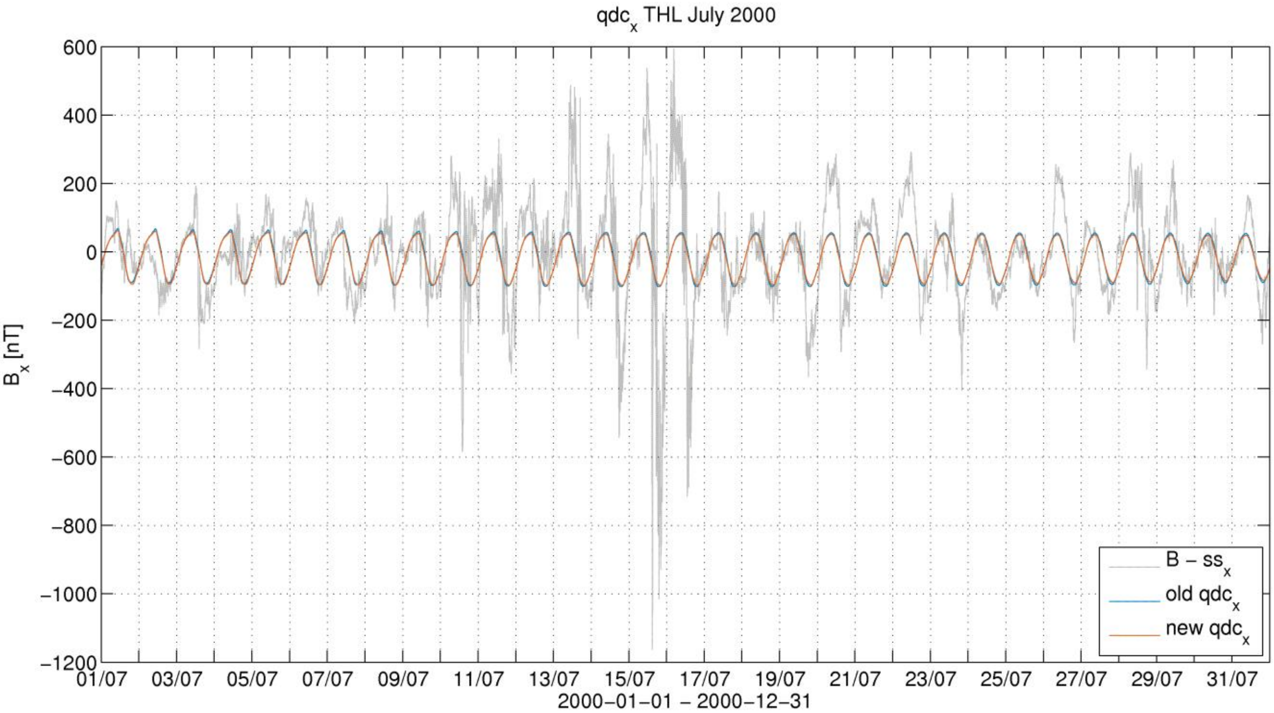

Figure 5-5 Quiet Daily Curve x-component for July 2000 using original and improved code

5.3 Impact on the definitive PC index

In general, as shown in the sections above, the more active periods, the larger the discrepancies between the results of the old and corrected implementation of the code, resulting in larger discrepancies in summer during years of high solar activity. Figure 5-6 shows the impact of the different changes on the definitive PC index for the northern hemisphere (PCN) in year 2000 and 2009 respectively. It is seen that in years of low solar activity (here 2009), the resulting differences due to the changes in the new algorithm are very small, of the order of ±0.1 mV/m. The largest discrepancies, above ±1 mV/m, occur around New Year in years of high solar activity (here 2000), and are mostly due to the changes in the SS caused by the added one month data overlap to the processing (see Section 5.1). The changes introduced by increasing the significant digits and correcting the Actual Day date are very small and consistent, as expected. The correction to the QDC routine, where the *moving* method has been added, only shows differences up to ±0.3 mV/m during summer months in years of high solar activity. Figure 5-7 presents the discrepancies in the entire definitive PCN dataset from 1975 until 2014. In general the deviations between the original and new codes are very small, of the order of ±0.1 mV/m. It is seen that there is a strong correlation between the higher discrepancies and the solar activity (green line), as described above.

By implementing the improved algorithm described in this document, the general changes in the definitive PCN index are expected to be in the order of up to ±0.1 mV/m, with very few larger changes between ±0.3 mV/m and ±2 mV/m during summer months and around New Year.

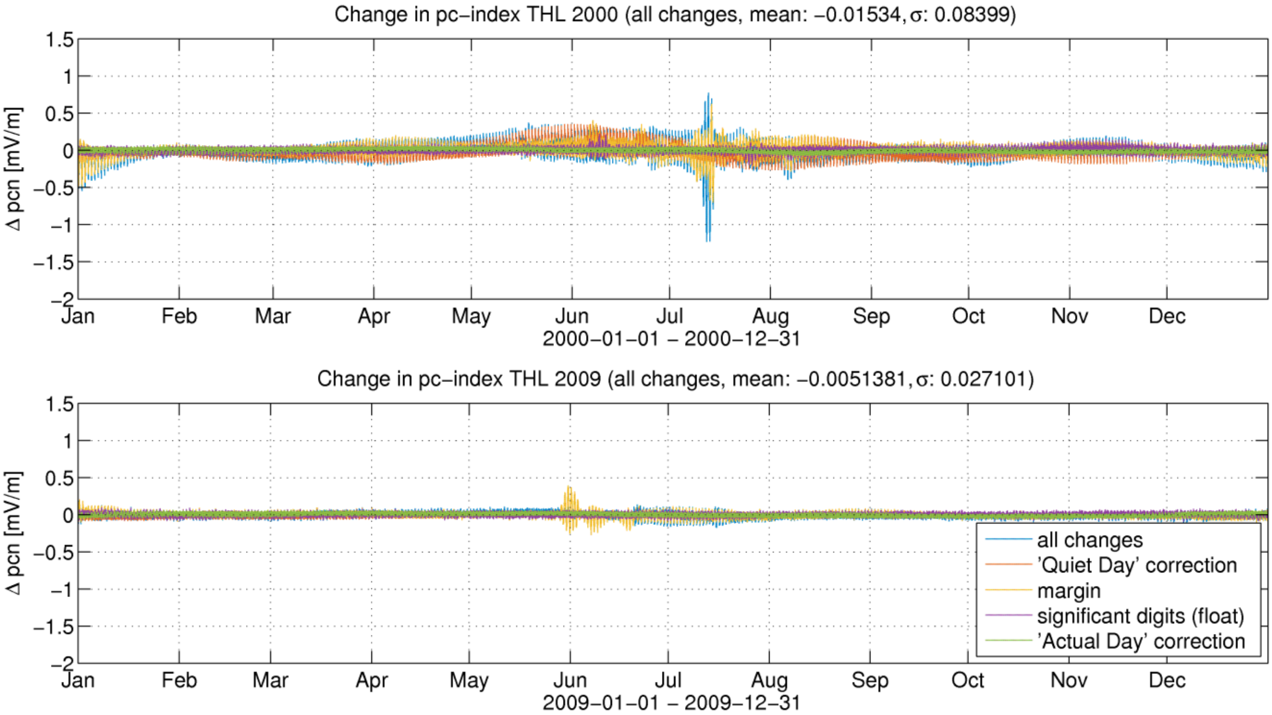

Figure 5-6 Difference in definitive PC index caused by changes in improved code for 2000 and 2009

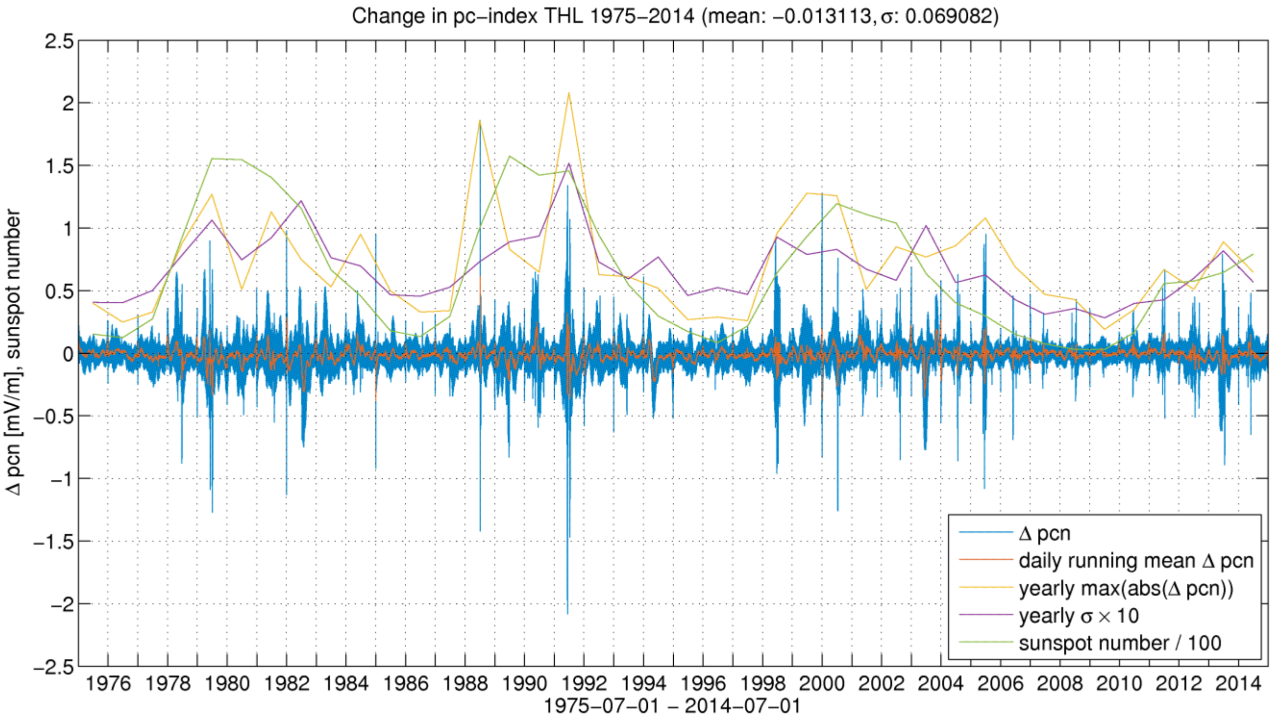

Figure 5-7 Changes in definitive PC index from 1975 until 2014

6 Appendix A

6.1 Definitive_PC_Index.m

The main program where most of the calculations are made. This program calls the functions `append_db.m` (see 6.2) and `q_day.m` (see 6.3), moreover `db_config.pc` and `coeff.mat` have to be present (see [AD-2])

```
%% Calculation of definitive PC index
% Compute the definitive PC index for the northern hemisphere (thl) or the
% southern hemmisphere (vos).
% Results are written to a SQL database defined in db_config.pc
%
% SQL table information
% +-----------+-------------+------+-------------------+------------------------+
% | COLUMN    | TYPE        | NULL | DEFAULT           | COMMENTS               |
% +-----------+-------------+------+-------------------+------------------------+
% | id        | int(11)     | No   |                   | SQL auto increment id  |
% +-----------+-------------+------+-------------------+------------------------+
% | timestamp | timestamp   | No   | CURRENT_TIMESTAMP | SQL entry time         |
% +-----------+-------------+------+-------------------+------------------------+
% | time      | datetime    | No   |                   | time of the measurement |
% +-----------+-------------+------+-------------------+------------------------+
% | x         | float       | Yes  | NULL              | X component            |
% +-----------+-------------+------+-------------------+------------------------+
% | y         | float       | Yes  | NULL              | Y component            |
% +-----------+-------------+------+-------------------+------------------------+
% | ss_x      | float       | Yes  | NULL              | SS(x)                  |
% +-----------+-------------+------+-------------------+------------------------+
% | ss_y      | float       | Yes  | NULL              | SS(y)                  |
% +-----------+-------------+------+-------------------+------------------------+
% | qdc_x     | float       | Yes  | NULL              | QDC(x)                 |
% +-----------+-------------+------+-------------------+------------------------+
% | qdc_y     | float       | Yes  | NULL              | QDC(y)                 |
% +-----------+-------------+------+-------------------+------------------------+
% | pc        | float       | Yes  | NULL              | PC index               |
% +-----------+-------------+------+-------------------+------------------------+
%
% NOTE! To compute the definitive PC index a data overlap of +/- 1 month is
% needed
%
% Change list:
% - Entire code rewritten to be more time efficient
% - x, y, ss_x, ss_y, qdc_x and qdc_y changed from mediumint(9) to float
% - Data overlap added to avoid edge effects (+/- 1 month, can be modified)
% - The actual QDC is now calculate for a 30-day period instead of a 31-day
%   period
% - 'moving' added to smooth statement for "absolute deviations from trend"
%   computation in q_day subfunction
% - "Actual Day" is now saved to correct date - instead of the day after
% - All data are now written to the SQL database at the same time in
%   batches of 1000 lines (max limit of SQL INSERT-statement) to be more
%   time efficient
%
% $Revision: 2.0 $

% $Id: Definitive_PC_Index.m,v 2.0 2018/12/12 10:42:36 jnsnl Exp $

%% Input
% Select station [Qaanaaq (thl) or Vostok (vos)]
station = 'thl';

app = false; % Append available data on existing SQL table
if app
```

```
  % Time period (automatic append available data on existing SQL table)
  [t1, t2] = append_db(station);
else
  % Time periode (manual)
  t1 = 1975;      % start year
  t2 = 2014;      % end year
end%if app
% Data overlap +/- n days
overlap = 31; % 31 = +/- 1 month

% Path to definitive IAGA2002 files
iaga_dir = ['/nfs/r14/magobs/PC-index/Definitive_PC/data/' ...
            station '_archive/'];

% Load phi, alpha and beta coefficients
load('coeff.mat');

% Longitude of station
switch station
    case 'thl'
        lon = 291;
    case 'vos'
        lon = 106.9;
end%switch

% Read config file db_cong.pc
f = fopen('db_config.pc','r');
    serv = fgetl(f);
    log = fgetl(f);
    pass = fgetl(f);
    dbname  = fgetl(f);
    table = fgetl(f);
fclose(f);

%% Calculate definitive PC index for [station] in time period [t1] to [t2]
for year = t1:t2
    disp(['Computing definitive PC index (' station ') for ' num2str(year)])
    %% Initialize data structure
    data  = struct('t', [], 'x', [], 'y', [], 'ss_x', [], 'ss_y', [], ...
                   'qdc_x', [], 'qdc_y', [], 'pc', []);
    % Number of days (year +/- data overlap)
    ndays = sum(eomday(year,1:12)) + overlap*2;
    % Write time in minutes (year +/- data overlap) to data structure
    data.t = (datenum(year, 1, 1, 0, 0:ndays*1440-1, 0) - overlap)';
    % Fill the remaining data structure with NaN's
    fns = fieldnames(data);
    for fn = fns(2:end)' % [x, y, ss_x, ss_y, qdc_x, qdc_y, pc]
        data.(char(fn)) = NaN(length(data.t), 1);
    end%for fn

    %% Read definitive IAGA2002 data and add to data structure
    % Read every file in the year (+/- data overlap)
    for actual = datenum(year, 1, 1)-overlap:datenum(year, 12, 31)+overlap;
        % /[yyyy]/thl[yyyymmdd]dmin.min
        iaga_name = fullfile(iaga_dir, datestr(actual,'yyyy'), ...
                    [station, datestr(actual,'yyyymmdd'), 'dmin.min']);

        if ~exist(iaga_name, 'file')
          warning(['No ' iaga_name(end-18:end) '-file. Continue.'])
          continue
        end%if ~exist
        fid = fopen(iaga_name);
        % Find end of header (Not uniform across IAGA2002 files)
        % DATE       TIME         DOY     THLX      THLY      THLZ      THLF
        % 2011-12-31 23:59:00.000 365      2510.90   -3229.90  56246.10  88888.80
```

```
    while ~feof(fid)
        f = fgetl(fid);
        if strcmp(f(1:4), 'DATE')
          % Last header-line
          break
        end%if strcmp
    end%while ~feof
    % Read data form IAGA2002 file
    f = textscan(fid, '%s %s %d %f %f %f %f');
    fclose(fid);
    % Find index into data structure
    [~, i] = intersect(data.t, datenum([char(f{1}) char(f{2})], ...
                                       'yyyy-mm-ddHH:MM:SS.FFF'));
    % Insert x and y into data structure
    for j = 2:3
        f{j+2}((f{j+2} >= 88888)) = NaN;
        data.(char(fns(j)))(i) = f{j+2};
    end%for j
end%for actual

%% Calculate SS (sector structure)
% Calculate ss_x and ss_y
for fn = fns(2:3)' % [x, y]
    % Calculate daily median value
    day_med = nanmedian(reshape(data.(char(fn)), [1440 ndays]));
    % Smooth medians
    w = smooth(day_med, 7, 'moving'); % 7 days
    w = smooth(w, 7, 'rloess');

    % Interpolation to minutes and add to data structure
    data.(['ss_' char(fn)]) = interp1(1:ndays, w, ...
                                  1:1/1440:ndays+1-1/1440, ...
                                  'spline', 'extrap')';
end%for fn

%% Calculate QDC (Actual Day)
% Shift 30-day period by 10 days at a time to calculation a series of
% actual QD:
for fn = fns(2:3)' %[x, y]
    for start_dt = datenum(year, 1, 1)-overlap:10:datenum(year, 12, 31)
        % Calculate QDC for 30-days dataset and save it into
        % "Actual Day" in data structure
        end_dt = start_dt+30-1/1440;
        i = find(data.t >= start_dt & data.t <= end_dt);

        % Calculate COMP-SS
        by = data.(char(fn))(i) - data.(['ss_' char(fn)])(i);

        % Calculate QDC for an "Actual Day"
        [qday, ActDay] = q_day(by);

        % Write QDC for "Atcual Day" to data structure
        if ~isnan(ActDay) % If any
          j = data.t >= start_dt + ActDay - 1 & ...
              data.t <= start_dt + ActDay - 1/1440;
          data.(['qdc_' char(fn)])(j) = qday;
        end%if ~isnan
    end%for start_dt

    % Interpolation of QDC for every day from "Actual Day" array
    % "Actual Days"
    ActDays = find(~isnan(sum(reshape(data.(['qdc_' char(fn)]), ...
                                  [1440 ndays]))));

    % Construct 2D array of Actual QDC
```

```
    qdays = reshape(data.(['qdc_' char(fn)])(~isnan(data.(['qdc_' char(fn)]))), ...
                    [1440 length(ActDays)]);

    % Interpolation and smoothing QDC for every day
    interp_arr = NaN(1440, ndays);
    for i = 1:1440
        interp_arr(i,:) = smooth(interp1(ActDays, qdays(i,:), 1:ndays, ...
                              'nearest', 'extrap'), 60, 'lowess');
    end

    % Make 1D array of the QDC, smooth and add to data structure
    l_arr = reshape(interp_arr, [1 ndays*1440]);
    data.(['qdc_' char(fn)]) = smooth(l_arr,120,'loess');
end%for fn

%% Caculate PC index
% Calculate disturbed part of the variations
dist_x = data.x - data.ss_x - data.qdc_x;
dist_y = data.y - data.ss_y - data.qdc_y;

% H Projection and PC calculation for the year (without data overlap)
i = 1+overlap*1440:(ndays-overlap)*1440;
[~, ~, ~, hour, minute, ~] = datevec(data.t(i));
% Minute in the year
min_in_year = i-overlap*1440;
% Calculate H_proj
UT = hour .* 15 + minute .* 0.25; % TIME to DEG
y = lon + coeff.f(min_in_year) + UT;
H_proj = dist_x(i).*sin(y.*pi./180)-dist_y(i).*cos(y.*pi./180);
% Calculate PC index and add to data structure
data.pc(i) = (H_proj - coeff.b(min_in_year))./coeff.a(min_in_year);

%% Write data structure to SQL database
mysql('open', serv, log, pass);
mysql(['use ' dbname]);
% Write data to SQL database in bulks of 1000 rows (max limit of SQL
% INSERT-statement) to save time
bulk = [overlap*1440:1000:(ndays-overlap)*1440, (ndays-overlap)*1440];
for i = 1:length(bulk)-1
    % Construct SQL query string
    str = [];
    for j = bulk(i)+1:bulk(i+1)
        str = [str '('];
        % Format time for SQL database
        str = [str '''' datestr(data.t(j),'yyyy-mm-dd HH:MM:SS') ''''];

        % Format data for SQL database
        for fn = fns(2:end)' % [x, y, ss_x, ss_y, qdc_x, qdc_y, pc]
            if ~isnan(data.(char(fn))(j))
              % Save data with two decimals in SQL database
              str = [str ', ' num2str(round(data.(char(fn))(j)*100)/100)];
            else
              % NaN is set to NULL
              str = [str ', NULL'];
            end%if ~isnan
        end%for fn
        str = [str '), '];
    end%for j
    % SQL query
    query = ['INSERT INTO ' table ' (time, x, y, ss_x, ss_y, qdc_x, qdc_y, pc) ' ...
             'VALUES ' str(1:end-2) ' '...
             'ON DUPLICATE KEY UPDATE x = VALUES(x), y = VALUES(y), ' ...
                    'ss_x = VALUES(ss_x), ss_y = VALUES(ss_y), ' ...
                    'qdc_x = VALUES(qdc_x), qdc_y = VALUES(qdc_y), ' ...
                    'pc = VALUES(pc);'];
```

```
        mysql(query);
    end%for i
    mysql('close');
end%for year
```

6.2 append_db.m

This sub function is called by Definitiv_PC_Index.m if the app-variable is true. The function will determine the computational time period based on available definitive WDC data and the available data in the SQL database.

```
function [t1, t2] = append_db(station)
% Determine time period to append available data to existing SQL table
% t1   year after newest year in existing SQL table
% t2   year of latest available definitive data

% Read config file db_cong.pc
f = fopen('db_config.pc','r');
    serv = fgetl(f);
    log = fgetl(f);
    pass = fgetl(f);
    dbname  = fgetl(f);
    table = fgetl(f);
fclose(f);

% Get latest year from db
mysql('open', serv,log,pass);
mysql(['use ' dbname]);
t1 = mysql(['select time from ' table ' where id=(select max(id) from ' table
');']);
mysql('close');
% Year after newest year in existing SQL table
t1 = str2num(datestr(t1, 'yyyy'));

% Available data
station_dir = ['/nfs/r14/magobs/PC-index/Definitive_PC/data/' station '_ar-
chive/'];
year_folders = dir(station_dir);
for i = 1:length(year_folders)
    t2 = str2num(year_folders(end-(i)).name);
    nfiles = dir(fullfile(station_dir, num2str(t2), '*.min'));
    if(t2 > t1 && length(nfiles) >= 365)
      break
    elseif(t2 <= t1)
      error(['No new data available. Newest data: ' num2str(t2)])
    end%if
end%for i
```

6.3 q_day.m

This sub function is called by Definitiv_PC_Index.m and calculates the Quiet Day Curve and the Actual Quiet Day for a 30 day period.

```
%% QDC calculation
% for actual day
function [qday, ActDay] = q_day(arr)
% [qday, ActDay] = q_day(arr)
```

```
%
% The function calculates actual Qiuet Day Curve (QDC) for
% INPUT array (arr) of data.
% arr - is 1-minutes values for 30 days = 43200 elements
% OUTPUT
% qday - array of 1400 elements (1-day) QDC values
% ActDay - actual day in 30 days array
%
% Change list:
% - 'moving' added to smooth statement for "absolute deviations from trend"
% computation
%
% $Revision: 2.0 $

% $Id: q_day.m,v 2.0 2018/12/12 10:47:45 jnsnl Exp $

% Absolute deviations from trend
arr_d = abs(arr - smooth(arr, 120, 'moving'));

% Absolute gradients
arr_g = abs(gradient(arr));

% Shift 61 min period by 1 min at a time and find the maximum deviations
% from trend and maximum gradients
for i = 31:length(arr)-30
    % Maximum deviations from trend in the 61 min interval
    max_ds(i-30,1) = max(abs(arr_d(i-30:i+30)));
    % Maximum gradients in the 61 min interval
    max_g(i-30,1) = max(abs(arr_g(i-30:i+30)));
end%for i

% Initialize test parameters
%Quiet value limit
divq = 0;
% Selected quiet data day-matrix
day = [];
% Selected quiet time vector
act(1:50000) = NaN;
ac = 1;
% Quiet day covered
p = 0;
% Quiet day exists
flag = true;

% Find quiet data covering an entire "day" by iterating through quiet
% limits from 2 nT to <40 nT in steps of 2 nT until a sufficient number of
% 1-min quiet segments is found
while min(p) < 120 % Find at least 120 quiet samples within 2 hour periodes
    % Increase quiet limit by 2 nT
    divq = divq + 2;
    if divq >= 40 % If quiet limit reaches 40 nT
        % Quiet day not covered
        flag = false;
        break
    end%if actual
    clear p;

    % Find data within quiet limit
    q = NaN(length(arr), 1);
```

```
    i = find(max_ds < divq & max_g < divq);
    q(i+30) = arr(i+30);

    % Append quiet data to quiet data day-matrix
    day = [day, reshape(q, [1440 length(arr)/1440])];

    % Quiet minutes within quiet limit
    act(ac:ac+length(i)-1) = i+30;
    ac = ac + length(i);

    % Find number of quiet data-points within 2 hour periods to ensure that
    % quiet day is covered. At least 120 quiet samples within 2 hours
    for i = 1:1320
        p(i) = length(find(~isnan(day(i:i+120,:))));
    end%for i
end%while

if flag % Quiet day covered
    % Repeat quiet day and calculate mean for every quiet minute
    qqq = nanmean([day; day; day]');
    % Smooth Quiet Daily Curve
    qqq_s  = smooth(qqq, 240, 'moving');
    qqq_s = smooth(qqq_s, 240, 'rlowess');
    % Quiet Daily Curve covering one day
    qday = qqq_s(1440:2880-1);
    % Calculate Actual Day
    ActDay = round(nanmedian(act)/1440);
else
    % No Quiet Daily Curve found
    qday(1:1440) = NaN;
    % No Actual quiet day found
    ActDay = NaN;
end
```