

УДК 2.3.3

Разработка и применение программы для автоматического нахождения невязки при пространственном контроле режимной метеорологической информации

А. И. Честнов

Российский государственный гидрометеорологический университет, Россия, 192007, Санкт-Петербург, Воронежская ул., 79

Для цитирования: Честнов А. И. Разработка и применение программы для автоматического нахождения невязки при пространственном контроле режимной метеорологической информации // Известия Петербургского государственного университета путей сообщения. СПб.: ПГУПС, 2024. Т. 21, вып. 1. С. 32–46. DOI: 10.20295/1815-588X-2024-01-32-46

Аннотация

Цель: разработать и применить программное обеспечение для автоматического нахождения невязок в автоматизированных метеорологических станциях с целью выявления сомнительных и недостоверных данных без непосредственного ручного контроля на языке Fortran с применением запросов языка программирования PostgreSQL. Написать блок программы для обработки данных из архива базы данных метеорологических станций единой сети. Установив соединение, с помощью запросов обеспечить поступление данных усредненных значений 10-минутных метеорологических сводок по специальной методике обработки. Составить подпрограмму для решения уравнения методом наименьших квадратов матричным методом, сопоставить для проверки результатов ее с методом линейной регрессии в стороннем приложении. Обеспечить должную запись данных перед отправкой их в базу данных, подготовить необходимую таблицу для корректной демонстрации данных и удобства пользователя в их использовании. Сформулировать корректные запросы для отправки в базу данных результатов нахождения невязки. Сформулировать запросы для создания необходимой таблицы, сформулировать запросы и выявить условия для реализации программы для ее более гибкого функционирования, то есть возможность обработки данных и нахождения невязки в случае неежедневной работы программы. **Методы:** методика схожа с принципами рекомендаций по анализу результатов пространственного контроля режимной метеорологической информации Главной геофизической обсерватории им. А.И. Воейкова. Методы включают перевод координат в декартову систему и реализацию решения систем уравнения методом наименьших квадратов. **Результаты:** создана программа, способная работать без участия оператора, производящая автоматический запуск, обработку данных и запись данных для дальнейшего хранения. **Практическая важность:** программа позволяет получать данные оперативно благодаря быстрой скорости обработки без ошибок, вызванных человеческим фактором.

Ключевые слова: невязка, автоматическая обработка, метеостанции, язык программирования Fortran, PostgreSQL.

Метеорологическая информация играет важную роль в обеспечении судоходства морского и речного транспорта, дорожной инфраструктуры, как в автомобильной, так и в железнодорожной. Исправное функцио-

нирование автоматических метеорологических станций позволяет удовлетворить нужды гидрометеорологического обеспечения служб, нуждающихся в нем без привлечения большого количества людей, что снижает

потребность людских ресурсов и помогает экономить денежные [1–2].

Помимо этого, информация от автоматических метеорологических станций поступает намного более оперативно, чем от неавтоматизированных, на которых измерение, запись и отправка данных происходят дежурным инженером-синоптиком или метеорологом. Только на автоматических станциях возможна постоянная запись и отправка десятиминутных сводок, которые используются для оперативных прогнозов погоды, корректирующие долгосрочные [3–4].

Оперативные прогнозы незаменимы для автомобильной дорожной инфраструктуры и в авиационной метеорологии. Они позволяют своевременно предупреждать опасные явления погоды, такие как обледенение; это позволит оперативно известить водителей на автомагистралях через электронное табло или отменить рейс в аэропорту для недопущения создания опасных ситуаций при перелете воздушного судна [5].

При огромном количестве преимуществ автоматические метеорологические станции имеют и ряд недостатков. Они выходят из строя в отличие от неавтоматизированных. Как правило, предупредить поломку или выход из строя прибора или всей станции очень тяжело. Если инженер-синоптик при очередном замере находит дефект или ущерб от чего-либо, нанесенный конструкции, и может оперативно устранить его, то поломки и проблемы автоматических метеостанций выявляются сильно позже при анализе данных, присылаемых оттуда.

Дефектов может быть много, как повреждение средства телекоммуникации, когда данные повреждаются при передаче или не доходят вовсе, так и неисправность датчиков, отправляющие неверную информацию.

При отсутствии данных или их малом количестве проблема выявляется быстро очень беглым анализом полноты данных. Достаточно сопоставить количество приходящих данных в небольшом интервале, например в суточном, с типом автоматической метеостанции. Так, в метеостанциях, присылающих 10-минутные сводки, должно за сутки приходиться 144 значения. При поступлении менее 86 значений, то есть 60% от общего числа возможных, прогноз по этой метеостанции становится недостоверным, а канал связи, приемник или передатчик показателей нуждается в ремонте [6].

Совсем другое дело обстоит с недостоверными показателями самих датчиков. На неавтоматизированных станциях синоптик самостоятельно может определить сбой в показателях при проверке метеоплощадки или даже при снятии показателей, если дефект прибора виден сразу. В таком случае производится ремонт на месте или замена. Такие метеостанции часто считаются контрольными: показания автоматических сверяются с ними при расположении в одном радиусе 150 км [7].

Несмотря на относительную близость расположений, показания не должны полностью совпадать друг с другом, поэтому выявление сомнительных и недостоверных данных требует комплексного и сложного анализа.

Данный анализ результатов был разработан Федеральной службой России по гидрометеорологии и мониторингу окружающей среды специалистами Главной геофизической обсерватории им. А. И. Воейкова (далее — ГГО) и издан в 1993 году Санкт-Петербургским «Гидрометеоиздатом». Он изложен в рекомендациях по анализу пространственного контроля режимной метеорологической информации, являясь переработанным и дополненным

изданием «Методических указаний по проведению критического контроля результатов метеорологических наблюдений на сети станций» 1981 года [8].

Данные рекомендации предназначены для специалистов, занимающихся автоматизацией и осуществляющих контроль сетей метеостанций в целом.

Программа для автоматического нахождения невязки основана на методе пространственной интерполяции. Данная методика применима для района с нахождением от 3 метеостанций в схожих условиях рельефа местности. По данным метеостанций рассчитывается пространственное распределение величины в одной координатной плоскости. Происходит расчет невязки контролируемой станции с влияющими, что позволит определить, насколько сильно расходятся данные, учитывая ошибочные и сомнительные.

Первым действием происходит пересчет координат метеостанций в декартовы координаты с началом:

$$\begin{aligned} x_i &= 111,2(\lambda_0 - \lambda_i) \cos \frac{\varphi_0 + \varphi_i}{2}, \\ y_i &= 111,2(\varphi_0 - \varphi_i). \end{aligned} \quad (1)$$

где i — номер влияющей станции; λ_i и φ_i — соответственно широта и долгота станции; λ_0 и φ_0 — координаты контролируемой станции. 111,2 — расстояние между 1° оси Земли [9].

Второе действие — расчет пространственного распределения как полином первой степени от коэффициентов x и y .

$$f(x, y) = Ax + By + C. \quad (2)$$

Коэффициенты полинома A , B и C определяются методом наименьших квадратов

по измеренным значениям контролируемой величины P'_i на влияющих станциях.

$$\begin{aligned} S &= \sum_{i=1}^n \Delta_i^2 = \sum_{i=1}^n [f(x_i, y_i) - P'_i]^2 = \\ &= \sum_{i=1}^n (Ax_i + By_i + C - P'_i)^2_{\min} \end{aligned} \quad (3)$$

Условие будет выполнено, если отношение коэффициентов полинома будет равно 0.

$$\partial S / \partial A = 0; \quad \partial S / \partial B = 0; \quad \partial S / \partial C = 0. \quad (4)$$

То есть получится следующая система уравнений:

$$\left. \begin{aligned} A \sum_{i=1}^n x_i^2 + B \sum_{i=1}^n x_i y_i + C \sum_{i=1}^n x_i - \sum_{i=1}^n x_i P'_i &= 0 \\ A \sum_{i=1}^n x_i y_i + B \sum_{i=1}^n y_i^2 + C \sum_{i=1}^n y_i - \sum_{i=1}^n y_i P'_i &= 0 \\ A \sum_{i=1}^n x_i + B \sum_{i=1}^n y_i + Cn - \sum_{i=1}^n P'_i &= 0 \end{aligned} \right\} \quad (5)$$

Решение системы уравнений методом наименьших квадратов дает следующие значения для коэффициентов полинома:

$$D = \begin{vmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & n \end{vmatrix} \quad (6)$$

$$Dc = \begin{vmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i P_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i P_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & \sum_{i=1}^n P_i \end{vmatrix}$$

Значение полинома контролируемой величины равняется отношению двух полиномов друг к другу:

$$C = Dc / D. \quad (7)$$

И наконец, пространственная невязка равняется разности между полиномом контролируемой величины и среднего значения контролируемой станции за конкретный временной период:

$$\Delta = C - Pi. \quad (8)$$

Для корректной работы программы достаточно реализовать формулу пересчета координат в декартовы, а также нахождение двух полиномов после получения значения, выдаваемого на метеостанциях за определенный период.

Рекомендации ГГО рассчитаны для нахождения месячных невязок, однако принцип может распространяться и на декаду и даже на сутки, единственное — критерий достоверности данных будет отличаться [10].

Примером нахождения невязки послужит обработка данных белорусских автоматических метеостанций «Белгидромет» (рис. 1).

Необходимые данные с метеостанций хранятся на серверах в виде таблиц. Таблицы включают в себя идентификационный номер метеостанции из хранилища данных и идентификационный номер из таблицы координат. Для обработки в компиляторе их необходимо получить с помощью запроса на сервер, для этого подходит язык PostgreSQL, а вернутся данные в виде таблиц с помощью подпрограммы GetData АМО, разработанной специалистами Института радарной метеорологии [11].

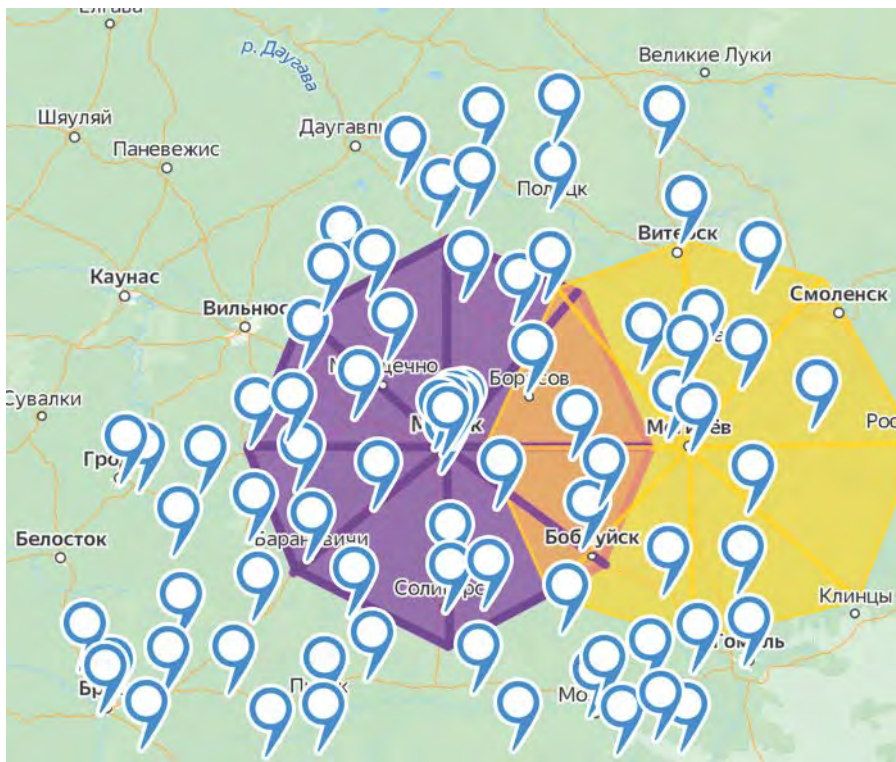


Рис. 1. Расположение метеостанций на территории Республики Беларусь

Запрос должен включать в себя нахождение нужного идентификационного номера и координаты, но при этом `id` необходимо заменить соответствующим из таблицы с данными.

В этом запросе идет преобразование данных и вывод их в отдельную переменную с соответствующими столбцами, чтобы колонка с координатами превратилась в две колонки, одна с долготой, вторая с широтой (табл. 1).

Субрутина (подпрограмма) `GetData АМО` возвращает многомерный массив структур,

ТАБЛИЦА 1. Результат запроса координат станций

id	locid	latitude	longitude
61	49	51.79667	30.2475
58	46	54.73333	28.35
62	50	52.11667	23.68583
63	51	52.25194	29.84056
64	52	55.82083	27.94
68	56	54.1	26.51667

размерность которого заранее задается в подпрограмме, в данном случае его размер равен 4 (рис. 2).

Далее из массива делается список. В языке Fortran, к сожалению, такого элемента нет, как в Python или в других языках, поэтому в качестве списка идет строка с элементами. Для ее получения необходимо преобразовать `g_Tuples` в более удобоваримый для обработки вид массива. Преобразование происходит с помощью функции `read`. Когда массив создан заранее и алоцирован, то есть для него выделена память на создание, запись и хранение, происходит чтение элементов из массива `g_Tuples` и присвоение им имени.

Чтение первого поля массива, его значения и размера, преобразование элемента в `integer`, то есть в целое число (идентификационный номер станции должен быть всегда целым, без цифр после запятой) и присваивание ему имени `id`. То же самое необходимо проделать с остальными элементами.

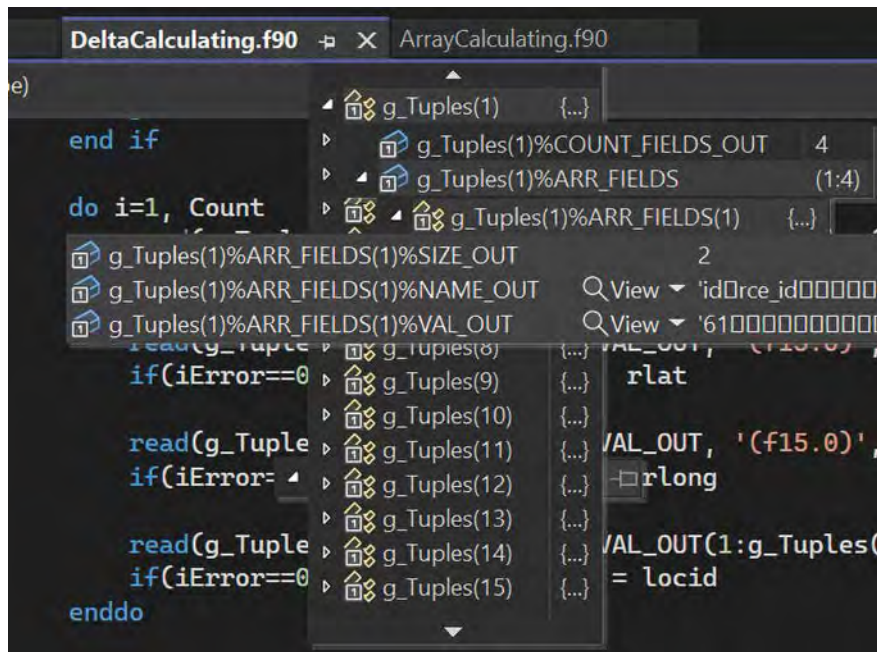


Рис. 2. Структура массива `g_Tuples`

Для последующей обработки значения широты и долготы преобразуются в вещественные, с 12 знаками после запятой без округления. Так как речь идет о координатах, необходима серьезная точность. Преобразуется точно так же, но вместо (i) прописывается (f15.0).

Когда массив готов, создается цикл, проходящий по всем элементам, с первого до последнего. Это позволит определить для каждой метеостанции соседние метеостанции, невязки которых необходимо найти и сравнить. Для определения соседних станций создается еще один SQL-запрос [12] (рис. 3).

```
select s.id, sl.id as locid, ST_Y(sl.point::geometry) as latitude,
ST_X(sl.point::geometry) as longitude
from info.source_location sl
inner join info.source s on sl.id = s.location_id
where ST_DWithin(sl.point, (select point from info.source_location sl2 where sl2.id = 49), 150000
```

Рис. 3. Скрипт для нахождения станций в радиусе 150 км

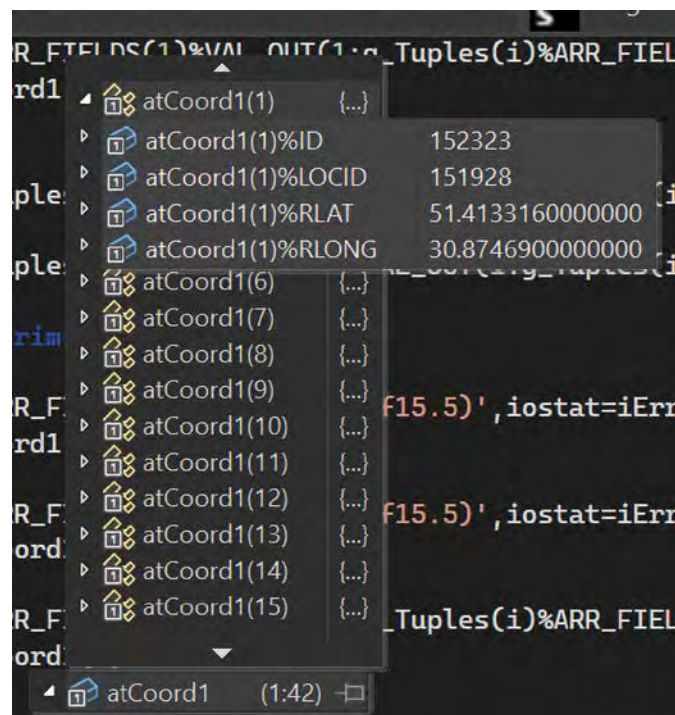


Рис. 4. Получившийся массив станций в радиусе 150 км, 42 метеостанции

В зависимости от периода, в котором происходит нахождение невязки, усреднение метеорологического параметра, в данном случае среднего значения температуры воздуха, команда будет немного отличаться [14].

Метеорологическое усреднение отличается от обычного математического дроблением на мелкие периоды. Сначала идет нахождение среднего значения для каждой 10-минутной сводки, из этих средних значений усредняется за один час, из них — суточное, из суточных — декадное, из декадных — месячное значения для каждой метеостанции [15].

Запрос для нахождения среднего значения температуры воздуха в архивах выглядит следующим образом (рис. 5).

Происходит усреднение по данным за один конкретный день, в данном случае за 30 августа 2023 года, с условием целостности данных не менее 60%. Это достигается последним действием, где заданы условия счетчику.

Аналогично сделаны запросы для трех декад месяца и месяца целиком, только там поставлены другие условия: у декад —

не меньше 864 значений наблюдений, у месяца — не менее 2600. Касаемо календарных декад возникает трудность в количестве дней, в третьей декаде может быть как 8 дней в феврале, так и 11 в октябре. Начало у третьей декады всегда одинаково, но конец разный, зато достоверно известно, что она ограничена началом следующего месяца, поэтому условие запроса должно исполняться тем, что диапазон дат строго меньше первого числа следующего месяца, то есть:

```
and time_obs > '2023-08-21' and time_obs
<= '2023-09-1'
```

После того как массивы со средними значениями были найдены, можно приступить к реализации формул. Первое действие — перевод всех координат, каждой метеостанции в декартову. Необходимо пройти циклом по каждой, поставив в формулу 1 [16] (рис. 6).

После перевода в систему все x и y суммируются для нахождения определителя матрицы, он находится через разницу треугольников умножения (рис. 7).

```
with AllBase as (select source_id, count(time_obs), avg(value_float[1]) as value_10min,
time_obs from meteo_point.surface_synoptic ss where value_float[1] is not null and source_id
in(152845,57,58,60,65,74,84,92,95,15588,88,15590,99,102,106,110,111,4378,5666,15647,15648,15646)
and time_obs > '2023-08-30' and time_obs <= '2023-08-31'
and measurand_id = 10 group by (source_id, time_obs)),
OneHour as (select source_id, avg(value_10min) as value_1h, (date_trunc('hour',time_obs)) as time_obs_H
from AllBase group by (source_id, time_obs_H) having count(time_obs) >= 4)
select avg(value_1h) as value, source_id
from OneHour group by (source_id) having count(value_1h) >= 24*0.6;
```

Рис. 5. Скрипт для нахождения среднего значения за определенный период

```
do m = 1, iStatInArea
  arXc(m) = (111.2 * (atAll_inout(i).rLat - arXc(m)) * cos((arYc(m) + atAll_inout(i).rLong)/2))
  arYc(m) = 111.2 * (atAll_inout(i).rLong - arYc(m))
end do
```

Рис. 6. Цикл для перевода данных в декартову систему

Для реализации нахождения определителя матрицы заранее создаются переменные, а потом подставляются в формулу (рис. 8).

Последними действиями остаются реализация формулы 7 и 8. Проверка осуществлялась вручную, но вместе с тем на Python была написана схожая программа для проверки, реализовывалась через функцию линейной регрессии библиотеки `sklearn.linear_model`.

Для реализации функции линейной регрессии был создан пустой массив для дальнейшего заполнения с именем `models` (рис. 9).

Во встроенную функцию линейной регрессии подается массив `X`, содержащий относительные координаты, то есть преобразованные в декартову систему и массив `Y`, в котором содержатся средние значения. После этого находится полином `C`, равный интерцепту, то есть перехвату значения `x`, когда на графике регрессии на своей оси координат оно равно 0. После этого из значения интерцепта вычитается среднее значение по станции. Оно совпадает с тем, что получилось в Fortran. Таким образом, выполнена

$$\begin{vmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & n \end{vmatrix} - \begin{vmatrix} \sum_{i=1}^n x_i^2 & \sum_{i=1}^n x_i y_i & \sum_{i=1}^n x_i \\ \sum_{i=1}^n x_i y_i & \sum_{i=1}^n y_i^2 & \sum_{i=1}^n y_i \\ \sum_{i=1}^n x_i & \sum_{i=1}^n y_i & n \end{vmatrix}$$

Рис. 7. Пример нахождения определителя матрицы

```

sum_x = sum(x)
sum_y = sum(y)
sum_x2 = sum(x**x)
sum_y2 = sum(y**y)
XiYi = sum(x**y)
Ni = size(atDayMean)-1
sum_FX = sum(x*atDayMean.rvalue)
sum_FY = sum(y*atDayMean.rvalue)

sum_F = sum_FALL - atDayMean(n)%rvalue
D = (sum_x2 * sum_y2 + Ni) + (sum_x * XiYi * sum_y) + (XiYi * sum_y * sum_x) - (sum_x2 * sum_y * sum_y) - (XiYi * XiYi * Ni)
Dc = (sum_x2 * sum_y2 + sum_F) + (sum_x * XiYi * sum_Fy) + (XiYi * sum_y * sum_Fx) - (sum_x * sum_y2 * sum_Fx) - (sum_x2 * sum_y * sum_Fy) - (XiYi * XiYi * sum_F)
C = Dc/D
Delt = (C - atDayMean(n)%rvalue)
atDayMean(n)%rDelta = Delt

```

Рис. 8. Реализация нахождения определителя матрицы

```

def getNumbers():
    models = []
    for i in range(0, len(X)):
        x_fit = np.delete(X[i], i, 0)
        y_fit = np.delete(Y, i, 0)
        models.append(LinearRegression().fit(x_fit, y_fit))
    return models
models = getNumbers()

ints = [ x.intercept_ for x in models]

```

Рис. 9. Реализация линейной регрессии в Python

двойная проверка, автоматическая и ручная, когда матричные уравнения решались самостоятельно (рис. 10).

Полученные значения записываются в отдельный массив, он необходим для дальнейшей записи значений в базу данных, которые также выполняются с помощью SQL-запроса.

Для написания его и передачи данных через подпрограмму GetData АМО необходимо создать строку, в которую будут укладываться нужные даты и значения невязок. Сначала создаются общие элементы строки, после они соединяются с переменными в зависимости от их значения. Перед записью данных создаются секции таблиц. Секционирование таблиц позволяет в дальнейшем оптимизировать про-

цесс поиска и фильтрации данных, что незаменимо для больших архивов (рис. 11).

Этот запрос создает секции в таблице, значения которых должны будут удовлетворять требованиям. Данные, которые пойдут на запись не в промежутке с 25 по 26 августа 2023 года, просто не будут занесены в нее.

Для передачи данных из компилятора используется подпрограмма GetData АМО с собранной строкой.

После создания секций последним действием является запись данных в саму таблицу, осуществляется через простую команду insert into (рис. 12).

Собирается строка аналогично, как строка для запроса создания секций; отправляется в базу данных тоже через GetData АМО.

Section	Value
atDayMean(1)%ID	61
atDayMean(1)%RVALUE	18.25403
atDayMean(1)%RLAT	51.7966670000000
atDayMean(1)%RLONG	30.2475000000000
atDayMean(1)%RDELTA	-0.294360429602428
atDayMean(7)	{...}

Рис. 10. Полученный массив после обработки

```
CREATE table if not exists [имя схемы]."[имя таблицы]_[имя секции]_2023_08-25"
PARTITION OF [имя таблицы]_[имя секции]
FOR VALUES FROM ('2023-08-25 00:00:00')
TO ('2023-08-26 00:00:00');
```

Рис. 11. Скрипт для создания секции таблицы

```
INSERT INTO [имя схемы]. [имя таблицы]
(source_id, measurand_id, processing_id, period_id, time_obs, time_cutoff, time_rec, value_float)
VALUES (61, 10, 14, 4, '2023-08-25 00:00:00',
'2023-10-20 10:21:55', timezone('UTC', NOW()), '{ -0.86}');
```

Рис. 12. Скрипт для добавления данных в таблицу

Source_id отвечает за идентификационный номер метеостанции; measurand_id — за измерение; processing_id — за манипуляцию с измерением, то есть в данном случае невязку; period_id — за временной период для невязки, день, месяц или календарная декада. Time_obs — время наблюдения метеорологического параметра, time_cutoff — время прекращения подачи сигнала, time_rec — время записи в таблицу, value_float — само значение (рис. 13).

Таким образом, схема одного цикла имеет следующий вид (рис. 14).

Программа рассчитана на полную автоматизацию и гибкость в использовании. Цикл проходит по всем метеостанциям в архиве, это значит, у каждой будет обнаружен радиус в 150 км, для каждой найдется невязка.

Последняя задача заключается в применении программы. Возможно, ее не удастся запускать каждый день для составления невязок месячных, декадных и дневных. В связи с этим была создана еще одна подпрограмма, определитель даты, базирующаяся на модуле MOD_TIME, тоже созданная специалистами

source_id	measurand_id	processing_id	period_id	time_obs	time_cutoff	time_rec
58	10	14	4	2023-05-02 00:00:00.000	2023-09-22 06:41:00.000	2023-09-22 06:41:00.000
58	10	2	4	2023-05-02 00:00:00.000	2023-09-22 06:41:00.000	2023-09-22 06:41:00.000
62	10	14	4	2023-05-02 00:00:00.000	2023-09-22 06:41:00.000	2023-09-22 06:41:00.000
62	10	2	4	2023-05-02 00:00:00.000	2023-09-22 06:41:00.000	2023-09-22 06:41:00.000
63	10	14	4	2023-05-02 00:00:00.000	2023-09-22 06:41:00.000	2023-09-22 06:41:00.000
63	10	2	4	2023-05-02 00:00:00.000	2023-09-22 06:41:00.000	2023-09-22 06:41:00.000

Рис. 13. Вид таблицы через СУБД

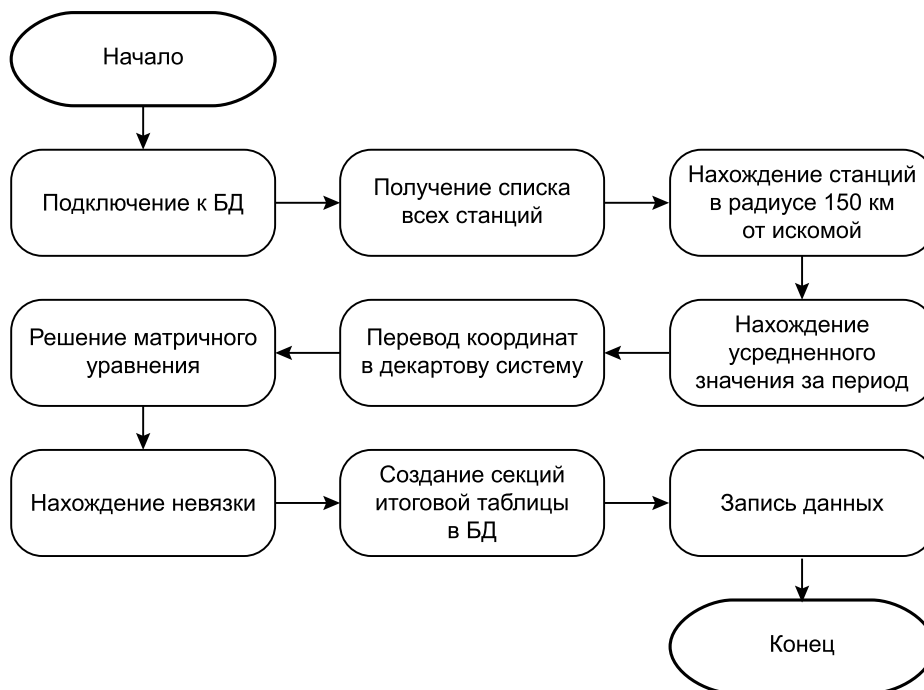


Рис. 14. Блок-схема одного цикла программы

ИРАМ. Модуль определяет время на компьютере, переводит его в UTC-Coordinated Universal Time и выдает дату. Было решено, что первого числа каждого месяца программа будет выполнять расчет по всему месяцу целиком, по трем календарным декадам и по суткам, в остальные дни выполняется расчет только суточных невязок [17].

Представить итоговые данные можно как в табличном виде (рис. 15), так и отобразив их на карте (рис. 16).

На подобной таблице прекрасно видно название станции и значение невязки. Зеленым цветом показывается допустимая

невязка, желтым — невязка слегка выходит за пределы нормы (0,5), красным цветом — серьезная невязка, выше 1 от предельно допустимого значения. Серый цвет означает отсутствие данных за период. Таблица служит для визуализации детального анализа, с помощью карты есть возможность произвести беглый анализ результатов и сделать выводы о количестве станций, нуждающихся в проверки данных.

Принцип отображения целиком идентичен таблице. У пользователя будет выбор слоев карт с определенными периодами: за месяц, сутки или декаду.

34	Ивье	
35	Имеретинка АМС	-3.05
36	Калинковичи	
37	Кепша АМС	7.84
38	Кичмай АМС	-0.35
39	Кличев	-0.18

Рис. 15. Таблица с данными невязки

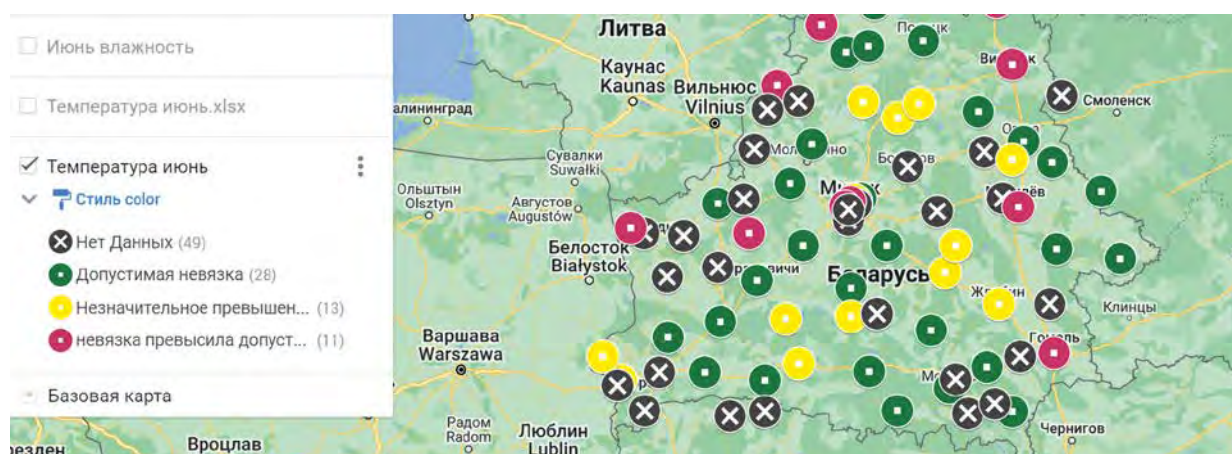


Рис. 16. Макет карты по визуализации данных станций

Выводы

В результате получилась программа, способная запускаться в любой день и полностью автоматизировать процесс нахождения важного метеорологического параметра. В интервале больше месяца программа работает довольно долго, более 10 минут, рекомендуется запускать ее чаще, чтобы интервалы были меньше. В любом случае предстоит еще процесс оптимизации. Программа имеет важное значение для метеорологов, синоптиков и климатологов, поскольку позволяет выявлять сомнительные и недостоверные данные без долгого и скрупулезного анализа данных, что ускорит их нахождение. Исключить человеческий фактор при выполнении данной задачи пока невозможно, потому что программа предоставляет данные для последующего анализа и пространственного контроля. Тем не менее исключение недостоверных данных позволит предупредить недостоверные и неоправданные прогнозы погоды, обледенение рек, гололеда и многих других явлений, критичных в судоходстве, автомобильной инфраструктуре и в других областях.

Библиографический список

1. Kolomeets L. I., Smyshlyaev S. P. Direct and indirect effects between thunderstorm activity, temperature and atmosphere composition on a regional scale: sensitive tests with WRF-CHEM. *Proceedings of MGO*. 2016. Vol. 585. P. 187–211.
2. Бочарников Н. В., Брылев Г. Б., Кузнецова Л. И. и др. Автоматизированные метеорологические радиолокационные комплексы «Метеоячейка». СПб.: Гидрометиздат, 2007. 236 с.
3. Fabry F. *Radar Meteorology: Principles and Practice*. Cambridge: Cambridge University Press, 2015.
4. Romps D. M., Seeley J., Vollaro D., et al. Projected increase in lightning strikes in the United States due to global warming // *Science*. 2014. V. 346. P. 851–854.
5. Kolomeets L. I., Smyshlyaev S. P. Regional and global lightning activity effect on the composition and properties of the upper troposphere/lower stratosphere. *Proceedings of SPIE The International Society for Optical Engineering*. 27. Сер. 27th International Symposium on Atmospheric and Ocean Optics, Atmospheric Physics, 2021. С. 1191671.
6. Гаврилова С. Ю., Иванова Т. А., Луцько Л. В. и др. О состоянии и функционировании автоматизированных метеорологической и актинометрической сетей в 2017 году. *Труды Главной геофизической обсерватории им. А. И. Воейкова*. 2018. № 588. С. 86–109.
7. Бочарников Н. В., Якимайнен Н. А. Использование данных метеорологических радиолокаторов при метеообеспечении авиации. В сб.: *Труды Международной конференции по авиационной и спутниковой метеорологии*. СПб.: Изд. РГГМУ, 2008. С. 145–148.
8. Лялюшкин А. С. Оптимизация стратегии сканирования доплеровского метеорологического радиолокатора. В сб.: *Труды Международной конференции по авиационной и спутниковой метеорологии*. СПб.: Изд. РГГМУ, 2008. С. 153–158.
9. Солонин А. С. Состояние и перспективы развития автоматизированных систем метеорологического обеспечения авиации. В сб.: *Труды Международной конференции по авиационной и спутниковой метеорологии*. СПб.: Изд. РГГМУ, 2008. С. 11–14.
10. Сикарев И. А., Честнов А. И., Абрамов В. М. Аспекты разработки и дальнейшие перспективы программы автоматической обработки спутниковых архивов гидрохимических данных на языке программирования Python // *Проблемы информационной безопасности. Компьютерные системы*. 2022. № 4 (52). С. 101–109.
11. Абрамов В. М., Сикарев И. А., Честнов А. И. и др. Автоматизированная обработка архивов метеорологических данных с помощью программы на языке Python // *Речной транспорт (XXI век)*. 2022. № 4 (104). С. 53–55.

12. Честнов А. И., Сикарев И. А., Абрамов В. М. Разработка программы для автоматической обработки данных направлений ветра // Информационные технологии и системы: управление, экономика, транспорт, право. 2022. № 4 (44). С. 117–121.

13. Базлова Т. А., Бочарников Н. В., Солонин А. С. Автоматизированная система метеорологического обеспечения службы содержания автомобильных дорог // Дороги России XXI века. 2002. № 1. С. 93–95.

14. Честнов А. И., Абрамов В. М., Голосовская В. А. и др. Создание макета системы по автоматизации обработки данных в строительной климатологии для проектирования объектов речного транспорта // Транспортное дело России. 2022. № 2. С. 212–216.

15. Сикарев И. А., Честнов А. И., Абрамов В. М. Аспекты разработки и дальнейшие перспективы... С. 101–109.

16. Честнов А. И., Сикарев И. А., Абрамов В. М. Автоматизация процесса по нахождению выбросов

в метеорологических архивах с помощью языка программирования Python. В сб.: Инновационные методы математики и физики в экологических и гидрометеорологических исследованиях. Сборник трудов Всероссийской научно-практической конференции; под ред. И. В. Зайцевой. СПб., 2023. С. 245–250.

17. Сикарев И. А., Абрамов В. М., Честнов А. И. и др. Разработка и применение программы для автоматического нахождения выбросов в метеорологических архивах // Морская радиоэлектроника. 2023. № 2 (84). С. 32–35.

Дата поступления: 15.12.2023

Решение о публикации: 28.01.2024

Контактная информация:

ЧЕСТНОВ Арсений Ильич — аспирант;
arsenij430@gmail.com

Development and application of a program for automatic finding of a discrepancy in the spatial control of regime meteorological information

A. I. Chestnov

Russian State Hydrometeorological University, 79, Voronezhskaya ul., Saint-Petersburg, 192007, Russia

For citation: *Chestnov A. I.* Development and application of a program for automatic finding of a discrepancy in the spatial control of regime meteorological information // Proceedings of Petersburg Transport University. 2024. Vol. 21, iss. 1. P. 32–46 (In Russian). DOI: 10.20295/1815-588X-2024-01-32-46

Abstract

Objective: to develop and apply software for automatic finding of discrepancies in automated meteorological stations in order to identify dubious and unreliable data without direct manual control in Fortran using PostgreSQL programming language queries. Write a program block for processing data from the archive of the database of meteorological stations of the unified network. Having provided a connection, with the help of requests, to ensure the receipt of data of the averaged values of ten-minute meteorological reports according to a special processing technique. Create a subroutine for solving the equation by the least squares matrix method, compare it to check the results with the linear regression method in a third-party application. Ensure proper recording of data before sending them to the database, prepare the necessary table for correct demonstration of data and user convenience in their use. Formulate correct queries to send the results of finding the discrepancy to the database. Formulate queries to create the necessary table, formulate queries and identify conditions for the implementation of the program for its more “flexible”

functioning, that is, the ability to process data and find a discrepancy in the case of non-daily operation of the program. **Methods:** the methodology is similar to the principles of recommendations for the analysis of the results of spatial monitoring of regime meteorological information of the Main Geophysical Observatory named after Voeikova A. I. Methods include the translation of coordinates into a Cartesian system and the implementation of the solution of equation systems by the least squares method. **Results:** a program has been created that can work without the participation of an operator, performing automatic startup, data processing and data recording for further storage. **Practical importance:** the program allows you to receive data quickly due to the fast processing speed without errors caused by the human factor.

Keywords: discrepancy, automatic processing, weather stations, Fortran programming language, PostgreSQL.

References

1. Kolomeets L. I., Smyshlyaev S. P. Direct and indirect effects between thunderstorm activity, temperature and atmosphere composition on a regional scale: sensitive tests with WRF-CHEM. Proceedings of MGO. 2016. V. 585. P. 187–211.
2. Bocharnikov N. V., Brylev G. B., Kuznecova L. I. i dr. Avtomatizirovannye meteorologicheskie radiolokacionnye komplekсы “Meteojachejka”. SPb.: Gidrometizdat, 2007. 236 s. (In Russian)
3. Fabry F. Radar. Meteorology: Principles and Practice. Cambridge: Cambridge University Press, 2015.
4. Romps D. M., Seeley J., Vollaro D., Molinari J. Projected increase in lightning strikes in the United States due to global warming // Science. 2014. V. 346. P. 851–854.
5. Kolomeets L. I., Smyshlyaev S. P. Regional and global lightning activity effect on the composition and properties of the upper troposphere/lower stratosphere. Proceedings of SPIE — The International Society for Optical Engineering. 27. Sep. 27th International Symposium on Atmospheric and Ocean Optics, Atmospheric Physics. 2021. C. 1191671.
6. Gavrilova S. Ju., Ivanova T. A., Luc’ko L. V. i dr. O sostojanii i funkcionirovanii avtomatizirovannyh meteorologicheskoy i aktinometricheskoy setej v 2017 godu. Trudy Glavnoj geofizicheskoy observatorii im. A. I. Voejkova. 2018. № 588. S. 86–109. (In Russian)
7. Bocharnikov N. V., Jakimajnen N. A. Ispol’zovanie dannyh meteorologicheskikh radiolokatorov pri meteo-
obespechenii aviacii. V sb.: Trudy Mezhdunarodnoj konferencii po aviacionnoj i sputnikovoj meteorologii. SPb.: Izd. RGGMU, 2008. S. 145–148. (In Russian)
8. Ljaljushkin A. S. Optimizacija strategii skanirovaniya doplerovskogo meteorologicheskogo radiolokatora. V sb.: Trudy Mezhdunarodnoj konferencii po aviacionnoj i sputnikovoj meteorologii. SPb.: Izd. RGGMU, 2008. S. 153–158. (In Russian)
9. Solonin A. S. Sostojanie i perspektivy razvitiya avtomatizirovannyh sistem meteorologicheskogo obespechenija aviacii. V sb.: Trudy Mezhdunarodnoj konferencii po aviacionnoj i sputnikovoj meteorologii. SPb.: Izd. RGGMU, 2008. S. 11–14. (In Russian)
10. Sikarev I. A., Chestnov A. I., Abramov V. M. Aspekty razrabotki i dal’nejshie perspektivy programmy avtomaticheskoy obrabotki sputnikovyh arhivov gidrohimičeskikh dannyh na jazyke programmirovanija Python // Problemy informacionnoj bezopasnosti. Komp’juternye sistemy. 2022. № 4 (52). S. 101–109. (In Russian)
11. Abramov V. M., Sikarev I. A., Chestnov A. I. i dr. Avtomatizirovannaja obrabotka arhivov meteorologicheskikh dannyh s pomoshh’ju programmy na jazyke Python // Rečnoj transport (HHI vek). 2022. № 4 (104). S. 53–55. (In Russian)
12. Chestnov A. I., Sikarev I. A., Abramov V. M. Razrabotka programmy dlja avtomaticheskoy obrabotki dannyh napravlenij vetra // Informacionnye tehnologii i sistemy: upravlenie, jekonomika, transport, pravo. 2022. № 4 (44). S. 117–121. (In Russian)

13. Bazlova T. A., Bocharnikov N. V., Solonin A. S. Avtomatizirovannaja sistema meteorologičeskogo obešpečenija služby soderžanija avtomobil'nyh dorog // Dorogi Rossii HXI veka. 2002. № 1. S. 93–95. (In Russian)
14. Chestnov A. I., Abramov V. M., Golosovskaja V. A. i dr. Sozdanie maketa sistemy po avtomatizacii obrabotki dannyh v stroitel'noj klimatologii dlja proektirovanija ob'ektov rechnogo transporta // Transportnoe delo Rossii. 2022. № 2. S. 212–216. (In Russian)
15. Sikarev I. A., Chestnov A. I., Abramov V. M. Aspekty razrabotki i dal'nejšie perspektivy... S. 101–109. (In Russian)
16. Chestnov A. I., Sikarev I. A., Abramov V. M. Avtomatizacija processa po nahozhdeniju vybrosov v meteorologičeskikh arhivah s pomoshh'ju jazyka programirovanija Python. V sb.: Innovacionnye metody matematiki i fiziki v jekologičeskikh i gidrometeorologičeskikh issledovanijah. Sbornik trudov Vserossijskoj nauchno-praktičeskoj konferencii. Pod redakciej I. V. Zajcevoj. SPb., 2023. S. 245–250. (In Russian)
17. Sikarev I. A., Abramov V. M., Chestnov A. I. i dr. Razrabotka i primenenie programmy dlja avtomatičeskogo nahozhdenija vybrosov v meteorologičeskikh arhivah // Morskaja radiojelektronika. 2023. № 2 (84). S. 32–35. (In Russian)

Received: 15.12.2023

Accepted: 28.01.2024

Author's information:

Arsenij Il. CHESTNOV — Postgraduate Student;
arsenij430@gmail.com