ПРИЛОЖЕНИЕ Б

**РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ**

**Библиотека открытого доступа DataReconcile для согласования неточных индустриальных данных за счет учета взаимосвязей между ними для нужд промышленных предприятий Индустрии 4.0 и целей Национальной технологической инициативы на языке**

**Python**

2025 год

**Б.1. НАЗНАЧЕНИЕ БИБЛИОТЕКИ**

Библиотека предназначена для решения задач **согласования неточных данных** (Data Reconcile) в условиях, когда входные данные содержат ошибки, неточности или противоречия. Она позволяет корректировать данные таким образом, чтобы они удовлетворяли заданным ограничениям, представленным в виде систем линейных и нелинейных уравнений и неравенств.

Библиотека помогает устранить противоречия в данных, минимизируя отклонения от исходных значений при соблюдении всех заданных ограничений. Это особенно полезно в задачах, где данные поступают из разных источников и могут содержать ошибки измерений или несоответствия.

**Применение в различных областях**:

Промышленность: согласование данных в системах управления технологическими процессами.

Экономика и финансы: обработка данных в моделях прогнозирования и анализа.

Наука и инженерия: решение задач оптимизации и калибровки моделей.

Энергетика: балансировка данных в энергосистемах.

Гибкость и масштабируемость:

Библиотека поддерживает работу с большими объемами данных и может быть интегрирована в различные программные системы.

Она предоставляет инструменты для настройки и адаптации под конкретные задачи пользователя.

**Примеры задач, где библиотека может быть полезна:**

Балансировка данных в химических процессах (например, согласование материальных и энергетических балансов).

Корректировка данных в системах мониторинга (например, устранение противоречий в показаниях датчиков).

Оптимизация данных в моделях прогнозирования (например, согласование исторических данных с прогнозными значениями).

Библиотека является мощным инструментом для анализа, корректировки и оптимизации данных в условиях наличия ограничений, что делает её незаменимой в задачах, требующих высокой точности и согласованности данных.

**Б.2. ОСНОВНЫЕ ВОЗМОЖНОСТИ**

Поддержка стационарных и динамических задач:

**Стационарные задачи**: работа с данными, которые не изменяются во времени (например, балансовые уравнения в статических системах).

**Динамические задачи**: обработка данных, которые изменяются во времени (например, временные ряды или процессы, описываемые дифференциальными уравнениями).

**Работа с ограничениями**:

Библиотека поддерживает ограничения в виде:

* + - **Линейных уравнений и неравенств** (например, **Ax**= **b** или **Ax**≤ **b**).
    - **Нелинейных уравнений и неравенств** (например, **F**(**x**) = **0** или **g**(**x**) ≤ **0**).

Это позволяет учитывать физические, экономические или другие законы, которым должны соответствовать данные.

**Минимизация отклонений**:

Библиотека минимизирует отклонения скорректированных данных от исходных, используя различные метрики.

**Б.3. СИСТЕМНЫЕ ТРЕБОВАНИЯ**

**Минимальные системные требования**

Минимальные требования позволяют запускать библиотеку и выполнять базовые задачи на компьютерах с ограниченными ресурсами. Однако производительность может быть низкой при работе с большими объемами данных или сложными задачами.

* **Операционная система**:
  + Windows 7/8/10/11 (64-битная) или новее.
  + macOS 10.13 (High Sierra) или новее.
  + Linux: Ubuntu 18.04 LTS или совместимые дистрибутивы.
* **Процессор (CPU)**:
  + Минимум: 2-ядерный процессор с тактовой частотой 1.8 ГГц (например, Intel Core i3 или аналогичный AMD).
* **Оперативная память (RAM)**:
  + Минимум: 4 ГБ (для небольших задач).
  + Рекомендуется: 8 ГБ для более комфортной работы.
* **Место на диске**:
  + Минимум: 500 МБ свободного места для установки Python и библиотек.
  + Дополнительное место для хранения данных и результатов.
* **Программное обеспечение**:
  + Python 3.7 или новее.
  + Установленные библиотеки:
    - numpy (для работы с массивами и матрицами).
    - scipy (для оптимизации и научных вычислений).
    - Дополнительные зависимости, указанные в документации библиотеки.

**Рекомендуемые системные требования**

Рекомендуемые требования обеспечивают высокую производительность при работе с большими объемами данных, сложными моделями и динамическими задачами.

* **Операционная система**:
  + Windows 10/11 (64-битная) или новее.
  + macOS 12 (Monterey) или новее.
  + Linux: Ubuntu 20.04 LTS или совместимые дистрибутивы.
* **Процессор (CPU)**:
  + Рекомендуется: 4-ядерный процессор с тактовой частотой 2.5 ГГц или выше (например, Intel Core i5/i7, AMD Ryzen 5/7).
* **Оперативная память (RAM)**:
  + Рекомендуется: 16 ГБ или больше (для работы с большими наборами данных и сложными задачами).
* **Место на диске**:
  + Рекомендуется: SSD с 1 ГБ свободного места для установки Python и библиотек.
  + Дополнительное место для хранения данных и результатов (в зависимости от объема данных).
* **Программное обеспечение**:
  + Python 3.9 или новее.
  + Установленные библиотеки:
    - numpy (последняя версия).
    - scipy (последняя версия).
    - Дополнительные библиотеки, такие как pandas (для работы с таблицами), matplotlib (для визуализации), joblib (для параллельных вычислений).
* **Графика**:
  + Дискретная видеокарта с 2 ГБ видеопамяти и поддержкой OpenGL 4.0 (например, NVIDIA GeForce GTX 1050 или выше).

**Особые рекомендации**

* **Для больших задач**:
  + Если вы работаете с очень большими наборами данных или сложными моделями, рекомендуется использовать серверные решения с многоядерными процессорами (например, 16+ ядер) и большим объемом оперативной памяти (32 ГБ и более).
  + Используйте библиотеки для параллельных вычислений, такие как joblib или dask.
* **Для динамических задач**:
  + Рекомендуется использовать SSD для быстрого чтения/записи данных, особенно при работе с временными рядами или большими объемами данных.
* **Для разработчиков**:
  + Убедитесь, что у вас установлены все необходимые пакеты и библиотеки, включая numpy, scipy и другие зависимые инструменты.
  + Используйте виртуальные окружения (venv или conda) для изоляции зависимостей.

**Проверка совместимости**

Перед установкой библиотеки убедитесь, что ваша система соответствует требованиям:

Проверьте версию Python и установленных библиотек (bash)

python --version

pip show numpy scipy

* Убедитесь, что ваш процессор и оперативная память соответствуют минимальным или рекомендуемым требованиям.
* Проверьте наличие свободного места на диске.

**Установка библиотек**

1. Установите Python с официального сайта: [python.org](https://www.python.org/).

Установите необходимые библиотеки с помощью pip (bash):

pip install numpy scipy

**Подготовка файлов библиотеки**

Здесь и далее в целях лаконичности используемая функция согласования данных обозначается условным именем \*LibFunctionName\*.

Создайте структуру папок:

Организуйте файлы библиотеки в отдельной папке. Например:

DataReconcile/

├── \_\_init\_\_.py

├── LibFunctionName1.py

├── LibFunctionName2.py

└── LibFunctionName3.py

└── …

* + Файл \_\_init\_\_.py необходим, чтобы Python распознал папку как пакет. Он может быть пустым или содержать код инициализации.

**Подключение библиотеки к проекту**

Скопируйте папку библиотеки в проект:

Поместите папку DataReconcile в папку вашего проекта. Например:

MyProject/

├── main.py

└── DataReconcile/

├── \_\_init\_\_.py

├── LibFunctionName1.py

├── LibFunctionName2.py

└── LibFunctionName3.py

└── …

**Импортируйте библиотеку в вашем проекте**:

В файле main.py (или другом файле проекта) используйте импорт:

from DataReconcile import LibFunctionName

**Б.4. ОСНОВНЫЕ ФУНКЦИИ И API**

**Б.4.1. Описание ключевых модулей**

Решаемая в рамках программной библиотеки задача относится к задачам повышения точности результатов совместных измерений, выполняемых на сложном объекте измерений, для которого предоставлена или получена в ходе эмпирических исследований математическая модель (возможно, не вполне точная). Снижение неопределенности получаемых результатов достигается за счет учета функциональных зависимостей между измеряемыми величинами, формализованных в рамках упомянутой математической модели. В качестве последней могут выступать как различные уравнения (алгебраические – линейные и нелинейные, дифференциальные и интегро-дифференциальные), а также наборы ограничений, вытекающие из условий решаемой измерительной задачи и условий функционирования и эксплуатации наблюдаемого объекта измерений (системы неравенств – односторонних и двусторонних). Решение задачи производится также при разном объеме известной информации, ключевое место среди которой занимают сведения о распределении погрешностей выполняемых измерений – как известно из математической статистики, при разных законах распределения случайных погрешностей разные числовые оценки для одного и того же параметра распределений будут являться эффективными (то есть будут обеспечивать наименьший статистический разброс от выборки к выборке). Данное обстоятельство указывает на обязательную необходимость привлечения сведений о распределении погрешностей (в том объеме, в котором она доступна). Решение задачи повышения точности (за счет согласования индустриальных данных) под перечисленными ограничениями представляет собой комплекс задач, решения которых имеет как общие черты, так и разнится в зависимости от типа ограничений. Эффективные вычислительные решения для разных значимых для измерительной практики ситуаций представляет собой основу настоящей библиотеки. На первом этапе основным типом рассматриваемых ограничений выступают математические модели, составленные из алгебраических уравнений (как линейных, так и нелинейных) при условии, что закон распределения случайных погрешностей отличается от нормального (возможно, несильно). Данный тип задачи должен включать в себя традиционные решения (в предположении гауссовости распределения погрешностей), так и новые, до сего момента не представленные в научной литературе по вопросу методы и подходы. Другим важным моментом является необходимость учета режима выполняемых измерений – статического (когда изменениями во времени значений измеряемых величин можно пренебречь без значимого снижения точности) или динамического (когда изменения сигналов измерительной информации во времени приводят к возникновению значимых составляющих погрешностей, отсутствие учета которых приводит к существенному снижению достоверности конечных результатов). Данная постановка задачи крайне важна для измерительной практики и предложена впервые в контексте задачи согласования индустриальных данных.

Рассматривая задачу согласования индустриальных данных целостно, следует отметить, что задача так или иначе сводится к условной оптимизационной задаче некоторой размерности и с тем или иным набором условий. Данное обстоятельство позволяет построить эффективную модель решения задачи согласования данных, выбрав для того или иного набора ограничений наиболее подходящий алгоритм оптимизации. Программная библиотека, реализующей разные методы согласования неточных данных друг с другом, также содержит в себе ряд автоматизированных инструментов.

На рисунке ниже представлена визуализация структуры библиотеки, а также входные и выходные данные.

Изображение выглядит как текст, снимок экрана, Шрифт, Параллельный

Автоматически созданное описание

Рис. – Структура библиотеки DataReconcile

В соответствии с приведенной на рисунке нумерацией, библиотека содержит:

1) программные средства, реализующие традиционные методы согласования индустриальных данных, которые содержатся и применяются в зарубежных коммерческих продуктах,

2) программные средства, реализующие непараметрические и условно непараметрические методы согласования индустриальных данных, позволяющие учитывать в рамках согласования существенные или малые отклонения действительного закона распределения погрешностей от нормального, применимые как в статическом режиме измерения, так и в динамическом,

3) программные представления и преобразования современных формализмов описания погрешности результатов измерений, таковых, чтобы результат преобразования/представления мог быть эффективно использован в качестве мера неопределенности неточных индустриальных данных при их согласовании по известной модели.

4) программные предварительной экспресс-оценки потенциального уточнения, достигаемого при математической обработке результатов выполняемых измерений за счет выполнения их согласования на основе известной априорной информации,

5) программные средства для обеспечения метрологического автосопровождения производимых вычислений [2-4], поскольку без него нет возможности оценить точность конечных результатов согласования, а также – при необходимости – произвести согласование моментов остановки выполняемых итерационных процедур с точностью исходных данных (то есть результатов выполненных измерений).

**Б.4.2. Доступные функции, их параметры и возвращаемые значения**

Презентуемая библиотека оформлена в виде независимых py-файлов, для работы с которыми требуется установленный интерпретатор Python версии 3.0 или новее, с установленными библиотеками numpy и Eigen. Ниже приведено описание процедур для не/полу/параметрического согласования данных и выполнения метрологического анализа потенциальных результатов согласования.

Для каждого параметра msrd\_data функции каждой функции справедливо следующее: в случае согласования результатов однократного совестного измерения нескольких взаимосвязанных величин, согласуемые результаты нужно передавать в формате вектор-столбца формата numpy array; если согласованию подлежит массив многократных совместных измерений, согласуемые значения нужно передавать в виде матрицы msrd\_data, где каждый столбец – набор однократного совместного измерения всех согласуемых величин (таким образом в *j*-ой строке предаваемой матрицы msrd\_data располагаются все результаты измерений *j*-ой измеряемой физической величины).

Презентуемая библиотека оформлена в виде независимых py-файлов, для работы с которым требуется интерпретатор языка программирования Python версии 3.0 или новее, включающий библиотеки numpy и scipy. Ниже приведено описание ключевых процедур для не/полу/параметрического согласования данных и выполнения метрологического анализа потенциальных результатов согласования.

Для каждого параметра msrd\_data функции каждой функции справедливо следующее: в случае согласования результатов однократного совестного измерения нескольких взаимосвязанных величин, согласуемые результаты нужно передавать в формате вектор-столбца; если согласованию подлежит массив многократных совместных измерений, согласуемые значения нужно передавать в виде матрицы msrd\_data, где каждый столбец – набор однократного совместного измерения всех согласуемых величин (таким образом в *j*-ой строке предаваемой матрицы msrd\_data располагаются все результаты измерений *j*-ой измеряемой физической величины).

*Процедура оценки степени потенциального уточнения совместных измерений за счет согласования*

|  |  |  |
| --- | --- | --- |
| estAccuracyIncreaseByDR | |  |
| Вызов | [accuracy\_increase\_ratio, variances\_of\_DR\_result] =  estAccuracyIncreaseByDR(dep\_func, ChosenMode, vals\_for\_reconc, no\_error\_params, errors) | |
| Описание | Функция выполняет приближенную оценку потенциального уточнения совместных измерений, достигаемого за счет учета известных функциональных взаимосвязей между измеряемыми величинами, на основе локальной линеаризации модели и метода декомпозиции алгоритма условной оптимизации. | |
| Аргументы | func – указатель на функцию, возвращающую значения уравнений, формализующих взаимосвязь между измеряемыми величинами.  ChosenMode – строка (string), позволяющей выбрать в каком режиме работает функция: по методу наименьших квадратов 'LS' или по обобщенному методу наименьших квадратов 'WLS',  vals\_for\_reconc – одномерный массив (ndarray), содержащий результаты совместных измерений, подлежащих уточнению;  no\_error\_params – одномерный массив (ndarray) значений параметров уравнений, описывающих зависимости между измеряемыми величинами.  errors – массив (ndarray), содержащий меры погрешностей согласуемых результатов измерений и параметров модели, используемой для согласования. | |
| Возвращаемое значение | accuracy\_increase\_ratio – одномерный массив (ndarray) оценок степени повышения точности, которое может быть достигнуто за счет процедуры согласования;  variances\_of\_DR\_result – одномерный массив (ndarray) оценок дисперсий результатов согласования. | |

***Функции согласования, учитывающего зависимости между взаимосвязанными величинами, выраженные в виде систем уравнений***

*Классическая процедура согласования совместных измерений в предположении, что случайные погрешности распределены нормально*

|  |  |
| --- | --- |
| DRparamEq |  |
| Вызов | [reconciled] =  DRparamEq(depend\_func, msrd\_data, error\_params, params) |
| Описание | Функция выполняет параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. |
| Аргументы | depend\_func – указатель на функцию (function), возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами,  msrd\_data – массив (ndarray), содержащий согласуемые результаты измерений,  error\_params – одномерный массив (ndarray), содержащий меры погрешности независимо полученных результатов измерений, подлежащих согласованию,  params – одномерный массив (ndarray) параметров модели взаимосвязей. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов параметрического согласования совместных измерений, выполненного при следующих допущениях: случайные погрешности результатов совместных измерений распределены нормально; согласуемые результаты измерений получены независимо, и, следовательно, корреляция между случайными погрешностями отсутствует |

*Процедура семи-непараметрического согласования с представлением неизвестного закона распределения погрешностей рядом Грама-Шарлье*

|  |  |
| --- | --- |
| DRsemiparamEq |  |
| Вызов | [reconciled] =  DRsemiparamEq(depend\_func, msrd\_data,  error\_params, alpha\_params, params) |
| Описание | Файл содержит одноименную вызываемую функцию, которая выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. |
| Аргументы | depend\_func – указатель на функцию (function), возвращающую значения левой части уравнений взаимосвязи между измеряемыми величинами;  msrd\_data – массив (ndarray), значения результатов совместного измерения величин, подлежащих согласованию;  error\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных меры погрешностей результатов измерений;  alpha\_params – двухмерный массив (ndarray), содержащий оценки среднеквадратического отклонения результатов измерений, коэффициентов асимметрии, коэффициентов и эксцесса случайных отклонений результатов измерений,  params – одномерный массив (ndarray), содержащий априорно заданные параметры модели, описывающей зависимости между согласуемыми величинами. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов семи-непараметрического согласования совместных измерений. |

*Процедура непараметрического согласования с ядерной аппроксимацией распределения случайных погрешностей*

|  |  |
| --- | --- |
| DRnonparamEq |  |
| Вызов | reconciled = DRnonparamEq(func, msrd\_data, model\_params, prior\_vars, bandwidths) |
| Описание | Функция выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. |
| Аргументы | func – указатель на функцию (function), возвращающую значения левой части уравнений вида **f***M*(**x**, **a**) = **0**, описывающих зависимости между измеряемыми величинами **x** и параметрами **a** модели;  msrd\_data – одномерный массив (ndarray), значения результатов однократного совместного измерения величин, подлежащих согласованию;  model\_params – одномерный массив (ndarray) параметров модели, описывающих зависимости между согласуемыми величинами, заданных точно;  prior\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных дисперсии результатов измерений;  bandwidth – одномерный массив (ndarray), содержащий регуляризирующее условие на результат ядерной аппроксимации неизвестного распределения случайных погрешностей согласуемых измерений. Данные значения являются ширинами окон ядерной аппроксимации. В качестве аппроксимирующего ядра использовано ядро Гаусса. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов непараметрического согласования результатов измерений взаимосвязанных величин. Предполагается, что измерения выполнялись независимо, и корреляция между случайными погрешностями отсутствует. В качестве процедуры идентификации неизвестного распределения случайных погрешностей согласуемых измерений применен метод ядерной аппроксимации ядром Гаусса. |

*Классическая процедура робастного согласования совместных измерений в предположении, что случайные погрешности распределены нормально*

|  |  |
| --- | --- |
| DRparamEqRobust |  |
| Вызов | [reconciled] =  DRparamEqRobust(depend\_func, msrd\_data,  error\_params, params\_) |
| Описание | Функция выполняет параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. |
| Аргументы | depend\_func – указатель на функцию (function), возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами,  msrd\_data – массив (ndarray), содержащий согласуемые результаты измерений,  error\_params – одномерный массив (ndarray), содержащий меры погрешности независимо полученных результатов измерений, подлежащих согласованию,  params – одномерный массив (ndarray) параметров модели взаимосвязей. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов параметрического согласования совместных измерений, выполненного при следующих допущениях: случайные погрешности результатов совместных измерений распределены нормально; согласуемые результаты измерений получены независимо, и, следовательно, корреляция между случайными погрешностями отсутствует |

*Процедура семи-непараметрического робастного согласования с представлением неизвестного закона распределения погрешностей рядом Грама-Шарлье*

|  |  |
| --- | --- |
| DRsemiparamEqRobust |  |
| Вызов | [reconciled] =  DRsemiparamEqReconcile(depend\_func, msrd\_data,  error\_params, alpha\_params, params) |
| Описание | Файл содержит одноименную вызываемую функцию, которая выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. |
| Аргументы | depend\_func – указатель на функцию (function), возвращающую значения левой части уравнений взаимосвязи между измеряемыми величинами;  msrd\_data – массив (ndarray), значения результатов совместного измерения величин, подлежащих согласованию;  error\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных меры погрешностей результатов измерений;  alpha\_params – двухмерный массив (ndarray), содержащий оценки среднеквадратического отклонения результатов измерений, коэффициентов асимметрии, коэффициентов и эксцесса случайных отклонений результатов измерений,  params – одномерный массив (ndarray), содержащий априорно заданные параметры модели, описывающей зависимости между согласуемыми величинами. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов семи-непараметрического согласования совместных измерений. |

*Процедура непараметрического робастного согласования с ядерной аппроксимацией распределения случайных погрешностей*

|  |  |  |
| --- | --- | --- |
| DRnonparamEqRobust | |  |
| Вызов | [reconciled] = DRnonparamEqRobust(func, msrd\_data, model\_params,  prior\_vars, bandwidths) | |
| Описание | Функция выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. | |
| Аргументы | func – указатель на функцию (function), возвращающую значения левой части уравнений вида **f***M*(**x**, **a**) = **0**, описывающих зависимости между измеряемыми величинами **x** и параметрами **a** модели;  msrd\_data – одномерный массив (ndarray), значения результатов однократного совместного измерения величин, подлежащих согласованию;  model\_params – одномерный массив (ndarray) параметров модели, описывающих зависимости между согласуемыми величинами, заданных точно;  prior\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных дисперсии результатов измерений;  bandwidth – одномерный массив (ndarray), содержащий регуляризирующее условие на результат ядерной аппроксимации неизвестного распределения случайных погрешностей согласуемых измерений. Данные значения являются ширинами окон ядерной аппроксимации. В качестве аппроксимирующего ядра использовано ядро Гаусса. | |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов непараметрического согласования результатов измерений взаимосвязанных величин. Предполагается, что измерения выполнялись независимо, и корреляция между случайными погрешностями отсутствует. В качестве процедуры идентификации неизвестного распределения случайных погрешностей согласуемых измерений применен метод ядерной аппроксимации ядром Гаусса. | |

***Функции согласования, учитывающего зависимости между взаимосвязанными величинами, выраженные в виде систем уравнений и неравенств***

*Классическая процедура согласования совместных измерений в предположении, что случайные погрешности распределены нормально*

|  |  |
| --- | --- |
| DRparamEqIneq |  |
| Вызов | [reconciled] =  DRparamEq(eqies\_model, ineqies\_model,  msrd\_data, error\_params, params) |
| Описание | Функция выполняет параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. Подлежат учету зависимости в виде равенств и неравенств. |
| Аргументы | eqies\_model – функция, возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами (размерность возвращаемого вектора соответствует размерности вектор-функции **f***M*),  ineqies\_model – функция, возвращающую результат вычисления левых частей системы неравенств вида **g***M*(**x**, **a**) ≥ **0**, представленных в формате равенств **g***M*(**x**, **a**) = **0** (в соответствии с методом множителей Лагранжа), описывающую взаимосвязи (ограничения) между согласуемыми величинами,  msrd\_data – массив (ndarray), содержащий согласуемые результаты измерений,  error\_params – одномерный массив (ndarray), содержащий меры погрешности независимо полученных результатов измерений, подлежащих согласованию,  params – одномерный массив (ndarray) параметров модели взаимосвязей. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов параметрического согласования совместных измерений, выполненного при следующих допущениях: случайные погрешности результатов совместных измерений распределены нормально; согласуемые результаты измерений получены независимо, и, следовательно, корреляция между случайными погрешностями отсутствует |

*Процедура семи-непараметрического согласования с представлением неизвестного закона распределения погрешностей рядом Грама-Шарлье*

|  |  |
| --- | --- |
| DRsemiparamEqIneq |  |
| Вызов | [reconciled] =  DRsemiparamEq(eqies\_model, ineqies\_model,  msrd\_data, error\_params, alpha\_params, params) |
| Описание | Файл содержит одноименную вызываемую функцию, которая выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. Подлежат учету зависимости в виде равенств и неравенств. |
| Аргументы | eqies\_model – функция, возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами (размерность возвращаемого вектора соответствует размерности вектор-функции **f***M*),  ineqies\_model – функция, возвращающую результат вычисления левых частей системы неравенств вида **g***M*(**x**, **a**) ≥ **0**, представленных в формате равенств **g***M*(**x**, **a**) = **0** (в соответствии с методом множителей Лагранжа), описывающую взаимосвязи (ограничения) между согласуемыми величинами,  msrd\_data – массив (ndarray), значения результатов совместного измерения величин, подлежащих согласованию;  error\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных меры погрешностей результатов измерений;  alpha\_params – двухмерный массив (ndarray), содержащий оценки среднеквадратического отклонения результатов измерений, коэффициентов асимметрии, коэффициентов и эксцесса случайных отклонений результатов измерений,  params – одномерный массив (ndarray), содержащий априорно заданные параметры модели, описывающей зависимости между согласуемыми величинами. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов семи-непараметрического согласования совместных измерений. |

*Процедура непараметрического согласования с ядерной аппроксимацией распределения случайных погрешностей*

|  |  |
| --- | --- |
| DRnonparamEqIneq |  |
| Вызов | [reconciled] = DRnonparamEqIneq (  eqies\_model, ineqies\_model,  msrd\_data, model\_params, prior\_vars, bandwidths) |
| Описание | Функция выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. Подлежат учету зависимости в виде равенств и неравенств. |
| Аргументы | eqies\_model – функция, возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами (размерность возвращаемого вектора соответствует размерности вектор-функции **f***M*),  ineqies\_model – функция, возвращающую результат вычисления левых частей системы неравенств вида **g***M*(**x**, **a**) ≥ **0**, представленных в формате равенств **g***M*(**x**, **a**) = **0** (в соответствии с методом множителей Лагранжа), описывающую взаимосвязи (ограничения) между согласуемыми величинами;  msrd\_data – одномерный массив (ndarray), значения результатов однократного совместного измерения величин, подлежащих согласованию;  model\_params – одномерный массив (ndarray) параметров модели, описывающих зависимости между согласуемыми величинами, заданных точно;  prior\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных дисперсии результатов измерений;  bandwidth – одномерный массив (ndarray), содержащий регуляризирующее условие на результат ядерной аппроксимации неизвестного распределения случайных погрешностей согласуемых измерений. Данные значения являются ширинами окон ядерной аппроксимации. В качестве аппроксимирующего ядра использовано ядро Гаусса. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов непараметрического согласования результатов измерений взаимосвязанных величин. Предполагается, что измерения выполнялись независимо, и корреляция между случайными погрешностями отсутствует. В качестве процедуры идентификации неизвестного распределения случайных погрешностей согласуемых измерений применен метод ядерной аппроксимации ядром Гаусса. |

*Классическая процедура робастного согласования совместных измерений в предположении, что случайные погрешности распределены нормально*

|  |  |
| --- | --- |
| DRparamEqIneqRobust |  |
| Вызов | [reconciled] =  DRparamEqIneqRobust(eqies\_model, ineqies\_model,  msrd\_data, error\_params, params\_) |
| Описание | Функция выполняет параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. Подлежат учету зависимости в виде равенств и неравенств. |
| Аргументы | eqies\_model – функция, возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами (размерность возвращаемого вектора соответствует размерности вектор-функции **f***M*),  ineqies\_model – функция, возвращающую результат вычисления левых частей системы неравенств вида **g***M*(**x**, **a**) ≥ **0**, представленных в формате равенств **g***M*(**x**, **a**) = **0** (в соответствии с методом множителей Лагранжа), описывающую взаимосвязи (ограничения) между согласуемыми величинами;  msrd\_data – массив (ndarray), содержащий согласуемые результаты измерений,  error\_params – одномерный массив (ndarray), содержащий меры погрешности независимо полученных результатов измерений, подлежащих согласованию,  params – одномерный массив (ndarray) параметров модели взаимосвязей. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов параметрического согласования совместных измерений, выполненного при следующих допущениях: случайные погрешности результатов совместных измерений распределены нормально; согласуемые результаты измерений получены независимо, и, следовательно, корреляция между случайными погрешностями отсутствует |

*Процедура семи-непараметрического робастного согласования с представлением неизвестного закона распределения погрешностей рядом Грама-Шарлье*

|  |  |
| --- | --- |
| DRsemiparamEqIneqRobust |  |
| Вызов | [reconciled] =  DRsemiparamEqIneqReconcile(eqies\_model,  ineqies\_model,  msrd\_data, error\_params,  alpha\_params, params) |
| Описание | Файл содержит одноименную вызываемую функцию, которая выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. Подлежат учету зависимости в виде равенств и неравенств. |
| Аргументы | eqies\_model – функция, возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами (размерность возвращаемого вектора соответствует размерности вектор-функции **f***M*),  ineqies\_model – функция, возвращающую результат вычисления левых частей системы неравенств вида **g***M*(**x**, **a**) ≥ **0**, представленных в формате равенств **g***M*(**x**, **a**) = **0** (в соответствии с методом множителей Лагранжа), описывающую взаимосвязи (ограничения) между согласуемыми величинами;  msrd\_data – массив (ndarray), значения результатов совместного измерения величин, подлежащих согласованию;  error\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных меры погрешностей результатов измерений;  alpha\_params – двухмерный массив (ndarray), содержащий оценки среднеквадратического отклонения результатов измерений, коэффициентов асимметрии, коэффициентов и эксцесса случайных отклонений результатов измерений,  params – одномерный массив (ndarray), содержащий априорно заданные параметры модели, описывающей зависимости между согласуемыми величинами. |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов семи-непараметрического согласования совместных измерений. |

*Процедура непараметрического робастного согласования с ядерной аппроксимацией распределения случайных погрешностей*

|  |  |  |
| --- | --- | --- |
| DRnonparamEqIneqRobust | |  |
| Вызов | [reconciled] = DRnonparamEqIneqRobust(  eqies\_model, ineqies\_model,  msrd\_data, model\_params,  prior\_vars, bandwidths) | |
| Описание | Функция выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. Подлежат учету зависимости в виде равенств и неравенств. | |
| Аргументы | eqies\_model – функция, возвращающую результат вычисления левых частей системы уравнений вида **f***M*(**x**, **a**) = **0**, описывающую взаимосвязи между согласуемыми величинами (размерность возвращаемого вектора соответствует размерности вектор-функции **f***M*),  ineqies\_model – функция, возвращающую результат вычисления левых частей системы неравенств вида **g***M*(**x**, **a**) ≥ **0**, представленных в формате равенств **g***M*(**x**, **a**) = **0** (в соответствии с методом множителей Лагранжа), описывающую взаимосвязи (ограничения) между согласуемыми величинами;  msrd\_data – одномерный массив (ndarray), значения результатов однократного совместного измерения величин, подлежащих согласованию;  model\_params – одномерный массив (ndarray) параметров модели, описывающих зависимости между согласуемыми величинами, заданных точно;  prior\_params – одномерный массив (ndarray), содержащий априорно заданные или оцененных дисперсии результатов измерений;  bandwidth – одномерный массив (ndarray), содержащий регуляризирующее условие на результат ядерной аппроксимации неизвестного распределения случайных погрешностей согласуемых измерений. Данные значения являются ширинами окон ядерной аппроксимации. В качестве аппроксимирующего ядра использовано ядро Гаусса. | |
| Возвращаемое значение | reconciled – одномерный массив (ndarray) результатов непараметрического согласования результатов измерений взаимосвязанных величин. Предполагается, что измерения выполнялись независимо, и корреляция между случайными погрешностями отсутствует. В качестве процедуры идентификации неизвестного распределения случайных погрешностей согласуемых измерений применен метод ядерной аппроксимации ядром Гаусса. | |

**Б.4.3. Примеры и особенности**

Приведённая в листинге 1 функция DRparamEq() предназначена для согласования измеренных данных с использованием метода, основанного на численном решении системы уравнений. Она использует модель зависимости между измеренными величинами, заданную функцией depend\_func, и априорные ошибки измерений для корректировки данных. Решается задача оптимальной оценки значений физических величин, известных с погрешностью, при этом априорно заданные взаимосвязи меду величинами формализованы в виде системы уравнений.

|  |
| --- |
| import numpy as np  from numpy import imag  from scipy.optimize import fsolve  def gauss\_system\_to\_solve(mu\_and\_l, depend\_func, msrd\_data, vars\_, params\_):  """ args=(depend\_func, msrd\_data, error\_params, params\_)  Solves for the gauss system to find zeros.  Parameters:  mu\_and\_l (ndarray): Array of unknown parameters (mu and lambda).  depend\_func (function): Dependency model function.  msrd\_data (ndarray): Measured data (n x xNum array).  vars\_ (ndarray): Error parameters, should not contain zeros.  params\_ (ndarray or list): Constant dependency model parameters.  Returns:  ndarray: A vector of values that need to be zero (mustbezeros).  """    if msrd\_data.ndim == 1:  xNum = msrd\_data.size  n = 1  elif msrd\_data.ndim == 2:  xNum = msrd\_data.shape[0]  n = msrd\_data.shape[1]  else:  raise ValueError(  "Only 2-dimensional arrays msrd\_data are supported.")    mustbezeros = np.zeros(xNum, dtype=np.complex128) # Preallocate zeros  for j in range(xNum):  if vars\_[j] == 0:  raise ValueError(  "None of the error parameters should be 0. Note that all constant "  "dependency model parameters need to be in the "  "'params\_' array.")  else:  # Imaginary perturbation for numerical derivative  imagx = np.zeros(xNum, dtype=np.complex128)  imagx[j] = 1j \* (mu\_and\_l[j] \* 10 \*\* (-100) + 10 \*\* (-101))    # Numerical derivative approximation using imaginary perturbation  df\_dx = imag(depend\_func(mu\_and\_l[:xNum] + imagx, params\_)) / imag(imagx[j])  # Update mustbezeros  mustbezeros[j] = (mu\_and\_l[j] / vars\_[j]  - np.mean(msrd\_data[j,:]) / vars\_[j]  + np.sum(mu\_and\_l[xNum:] \* df\_dx) / n)  # Evaluate the model residuals  model\_res = depend\_func(mu\_and\_l[:xNum], params\_)  mustbezeros = np.append(mustbezeros, model\_res)  return np.real(mustbezeros)  # -----------------------------------------------------------------------------  def GaussDataReconcil(depend\_func, msrd\_data, error\_params, params\_):  """  Reconciles measured data  Parameters:  depend\_func (function): Function describing relationships between measured quantities.  It can return either a single equation or a system of equations.  msrd\_data (ndarray): Vector of measured data.  error\_params (ndarray): A priori error variances/limits.  params\_ (ndarray): Known model parameters.  Returns:  ndarray: Reconciled measured data after applying the Alpha Gram-Charlier method.  """  data\_init\_points = np.mean(msrd\_data, axis=1)  # Step 1: Compute l (a zero vector of the same length as depend\_func's output)  l = np.zeros(len(depend\_func(data\_init\_points, params\_)))    # Step 2: Extend the initial guess (mu\_and\_l\_start) by appending l to msrd\_data  mu\_and\_l\_start = np.hstack((data\_init\_points, l))    # Solve the system of equations using fsolve  mu\_and\_l\_res = fsolve(gauss\_system\_to\_solve, mu\_and\_l\_start, args=(depend\_func, msrd\_data, error\_params, params\_))    # Step 4: Extract the reconciled measured data (excluding 'l' values)  reconciled\_ = mu\_and\_l\_res[:len(msrd\_data)]    return reconciled\_ |

Листинг 1. Функция DRparamEq() для согласования данных с использованием метода, основанного на численной оптимизации

**Пример использования функции**

Предположим, у нас есть набор измеренных данных, которые мы хотим согласовать с использованием модели зависимости.

Модель зависимости может быть, например, линейной функцией:

def linear\_model(mu, params):

"""

Линейная модель зависимости: y = a \* x + b.

Здесь params = [a, b].

"""

a, b = params

return a \* mu + b

**Подготовка данных**

Измеренные данные: 5 измерений, каждое измерение повторено 3 раза.

Априорные ошибки: заданы для каждого измерения.

Параметры модели: a = 2, b = 1.

import numpy as np

# Измеренные данные (5 измерений, каждое повторено 3 раза)

msrd\_data = np.array([

[1.1, 1.2, 1.3], # Измерение 1

[2.0, 2.1, 2.2], # Измерение 2

[3.0, 3.1, 3.2], # Измерение 3

[4.0, 4.1, 4.2], # Измерение 4

[5.0, 5.1, 5.2] # Измерение 5

])

# Априорные ошибки (дисперсии) для каждого измерения

error\_params = np.array([0.1, 0.1, 0.1, 0.1, 0.1])

# Параметры модели (a и b для линейной модели)

params\_ = np.array([2, 1])

**Вызов функции**

# Согласование данных

reconciled\_data = DRparamEq(linear\_model, msrd\_data, error\_params, params\_)

# Вывод результата

print("Согласованные данные:", reconciled\_data)

**Результат**

Функция вернет вектор согласованных данных, которые лучше соответствуют модели зависимости и учитывают погрешность измерений.

**Инициализация начальных значений**:

Начальные значения для согласованных данных (mu) берутся как средние значения измерений.

Начальные значения для множителей Лагранжа (l) устанавливаются в нули.

**Решение системы уравнений**:

Функция fsolve из библиотеки scipy.optimize используется для решения системы уравнений, которая формируется в функции gauss\_system\_to\_solve.

Система уравнений включает:

Уравнения для согласования данных с учетом априорных ошибок.

Уравнения, заданные моделью зависимости.

Функция gauss\_system\_to\_solve использует метод численного дифференцирования с мнимой возмущением для вычисления производных.

Если априорные ошибки (error\_params) содержат нули, функция вызовет ошибку, так как это приведет к делению на ноль.

Функция поддерживает одномерные и двумерные массивы для msrd\_data, то есть позволяет согласовывать однократные и многократные совместные измерения, соответственно.

Приведённая в листинге 2 функция DRparamEq() предназначена для согласования измеренных данных с использованием метода, основанного на численном решении системы уравнений. Она использует модель зависимости между измеренными величинами, заданную функцией depend\_func, и априорные ошибки измерений для корректировки данных. Решается задача оптимальной оценки значений физических величин, известных с погрешностью, при этом априорно заданные взаимосвязи меду величинами формализованы в виде системы уравнений.

|  |
| --- |
| import numpy as np  from numpy import imag  from scipy.optimize import fsolve  def gauss\_system\_to\_solve(mu\_and\_l, eqies\_model, ineqies\_model, msrd\_data, vars\_, params\_, xNum, n, eqNum, ineqNum):  """ args=(eqies\_model, msrd\_data, error\_params, params\_)  Solves for the gauss system to find zeros.  Parameters:  mu\_and\_l (ndarray): Array of unknown parameters (mu and lambda).  eqies\_model (function): Dependency model function.  msrd\_data (ndarray): Measured data (n x xNum array).  vars\_ (ndarray): Error parameters, should not contain zeros.  params\_ (ndarray or list): Constant dependency model parameters.  Returns:  ndarray: A vector of values that need to be zero (mustbezeros).  """    mustbezeros = np.zeros(xNum, dtype=np.complex128) # Preallocate zeros  for j in range(xNum):  if vars\_[j] == 0:  raise ValueError(  "None of the error parameters should be 0. Note that all constant "  "dependency model parameters need to be in the "  "'params\_' array.")  else:  # Imaginary perturbation for numerical derivative  imagx = np.zeros(xNum, dtype=np.complex128)  imagx[j] = 1j \* (mu\_and\_l[j] \* 10 \*\* (-100) + 10 \*\* (-101))    # Numerical derivative approximation using imaginary perturbation  df\_dx\_eqies = imag(eqies\_model(mu\_and\_l[:xNum] + imagx, params\_)) / imag(imagx[j])  df\_dx\_ineqies = imag(eqies\_model(mu\_and\_l[:xNum] + imagx, params\_)) / imag(imagx[j])  df\_dx = np.hstack((df\_dx\_eqies, df\_dx\_ineqies))  # Update mustbezeros  mustbezeros[j] = (mu\_and\_l[j] / vars\_[j]  - np.mean(msrd\_data[j,:]) / vars\_[j]  + np.sum(mu\_and\_l[xNum:(xNum+eqNum+ineqNum)] \* df\_dx) / n)  # Evaluate the model residuals  eqies\_model\_res = eqies\_model(mu\_and\_l[:xNum], params\_)  mustbezeros = np.append(mustbezeros, eqies\_model\_res)  ineqies\_model\_res = ineqies\_model(mu\_and\_l[:xNum], params\_)  mustbezeros = np.append(mustbezeros, ineqies\_model\_res)  mustbezeros = np.append(mustbezeros, ineqies\_model\_res\*mu\_and\_l[(xNum+eqNum):(xNum+eqNum+ineqNum)])  return np.real(mustbezeros)  # -----------------------------------------------------------------------------  def DRparamEqIneq(eqies\_model, ineqies\_model, msrd\_data, error\_params, params\_):  """  Reconciles measured data  Parameters:  depend\_func (function): Function describing relationships between measured quantities.  It can return either a single equation or a system of equations.  msrd\_data (ndarray): Vector of measured data.  error\_params (ndarray): A priori error variances/limits.  params\_ (ndarray): Known model parameters.  Returns:  ndarray: Reconciled measured data after applying the Alpha Gram-Charlier method.  """  if msrd\_data.ndim == 1:  xNum = msrd\_data.size  n = 1  elif msrd\_data.ndim == 2:  xNum = msrd\_data.shape[0]  n = msrd\_data.shape[1]  else:  raise ValueError(  "Only 2-dimensional arrays msrd\_data are supported.")  data\_init\_points = np.mean(msrd\_data, axis=1)  eqNum = len(eqies\_model(data\_init\_points, params\_))  ineqNum = len(ineqies\_model(data\_init\_points, params\_))  # Step 1: Compute l (a zero vector of the same length as eqies\_model's output)  l\_eqies = np.zeros(len(eqies\_model(data\_init\_points, params\_)))  l\_ineqies = np.zeros(len(ineqies\_model(data\_init\_points, params\_)))  l\_ineqies\_muly\_by\_l = np.zeros(len(ineqies\_model(data\_init\_points, params\_)))  l = np.hstack((l\_eqies, l\_ineqies, l\_ineqies\_muly\_by\_l))  # Step 2: Extend the initial guess (mu\_and\_l\_start) by appending l to msrd\_data  mu\_and\_l\_start = np.hstack((data\_init\_points, l))    # Solve the system of equations using fsolve  mu\_and\_l\_res = fsolve(gauss\_system\_to\_solve, mu\_and\_l\_start, args=(eqies\_model, ineqies\_model, msrd\_data, error\_params, params\_, xNum, n, eqNum, ineqNum))    # Step 4: Extract the reconciled measured data (excluding 'l' values)  reconciled\_ = mu\_and\_l\_res[:len(data\_init\_points)]    return reconciled\_ |

Листинг 2. Функция DRparamEq(), предназначенная для согласования измеренных данных с использованием метода, основанного на численном решении системы уравнений и неравенств по методу множителей Лагранжа.

**Пример использования функции DRparamEqIneq()**

Рассмотрим пример, где у нас есть набор измеренных данных, которые мы хотим согласовать с учетом уравнений и неравенств.

**Определение моделей уравнений и неравенств**

Уравнение: линейная зависимость.

Неравенство: ограничение на значения данных.

def eqies\_model(mu, params):

"""

Уравнение: линейная зависимость y = a \* x + b.

Здесь params = [a, b].

"""

a, b = params

return a \* mu + b

def ineqies\_model(mu, params):

"""

Неравенство: ограничение на значения данных (например, mu >= 0).

"""

return mu - 0 # mu >= 0

**Подготовка данных**

Измеренные данные: 3 измерения, каждое измерение повторено 2 раза.

Априорные ошибки: заданы для каждого измерения.

Параметры модели: a = 1, b = 0.

import numpy as np

# Измеренные данные (3 измерения, каждое повторено 2 раза)

msrd\_data = np.array([

[1.1, 1.2], # Измерение 1

[2.0, 2.1], # Измерение 2

[3.0, 3.1] # Измерение 3

])

# Априорные ошибки (дисперсии) для каждого измерения

error\_params = np.array([0.1, 0.1, 0.1])

# Параметры модели (a и b для линейной модели)

params\_ = np.array([1, 0])

**Вызов функции DRparamEqIneq()**

# Согласование данных

reconciled\_data = DRparamEqIneq(eqies\_model, ineqies\_model, msrd\_data, error\_params, params\_)

# Вывод результата

print("Согласованные данные:", reconciled\_data)

**Результат**

Функция вернет вектор согласованных данных, которые лучше соответствуют уравнениям и неравенствам, а также учитывают априорные ошибки измерений.

**Объяснение работы функции**

**Инициализация начальных значений**:

Начальные значения для согласованных данных (mu) берутся как средние значения измерений.

Начальные значения для множителей Лагранжа (l) устанавливаются в нули.

**Решение системы уравнений**:

Функция fsolve из библиотеки scipy.optimize используется для решения системы уравнений, которая формируется в функции gauss\_system\_to\_solve.

Система уравнений включает:

Уравнения для согласования данных с учетом априорных ошибок.

Уравнения, заданные моделью зависимости.

Неравенства, заданные моделью ограничений.

**Извлечение результата**:

После решения системы извлекаются согласованные данные (первые xNum элементов результата).

**Примечания**

Функция gauss\_system\_to\_solve использует метод численного дифференцирования с мнимой возмущением для вычисления производных.

Если априорные ошибки (error\_params) содержат нули, функция вызовет ошибку, так как это приведет к делению на ноль.

Функция поддерживает только двумерные массивы для msrd\_data.

Функция DRparamEqIneq() полезна для задач, где необходимо согласовать измеренные данные с учетом как уравнений, так и неравенств. Она может быть адаптирована для различных моделей и типов данных.

**Б.5. СПИСОК ФУНКЦИЙ И МЕТОДОВ**

|  |  |
| --- | --- |
| **estAccuracyIncreaseByDR()** | Функция выполняет приближенную оценку потенциального уточнения совместных измерений, достигаемого за счет учета известных функциональных взаимосвязей между измеряемыми величинами, на основе локальной линеаризации модели и метода декомпозиции алгоритма условной оптимизации. |
|  |  |
| **DRparamEq()** | Функция выполняет параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRsemiparamEq()** | Файл содержит одноименную вызываемую функцию, которая выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRnonparamEq()** | Функция выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRparamEqRobust()** | Функция выполняет робастное параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. |
| **DRsemiparamEqRobust()** | Файл содержит одноименную вызываемую функцию, которая выполняет робастное полу-непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. |
| **DRnonparamEqRobust()** | Функция выполняет робастное непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. |
|  |  |
| **DRparamEqIneq()** | Функция выполняет параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRsemiparamEqIneq()** | Файл содержит одноименную вызываемую функцию, которая выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRnonparamEqIneq()** | Функция выполняет непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRparamEqIneqRobust()** | Функция выполняет робастное параметрическое согласование совместно измеренных величин, закон распределения случайных погрешностей которых соответствует нормальному распределению вероятностей. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRsemiparamEqIneqRobust()** | Файл содержит одноименную вызываемую функцию, которая выполняет робастное полу-непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается проекционным методом с применением в качестве модели усеченного ряда Грамма-Шарлье. Подлежат учету зависимости в виде равенств и неравенств. |
| **DRnonparamEqIneqRobust()** | Функция выполняет робастное непараметрическое согласование совместно измеренных величин, закон распределения которых оценивается методом ядерной аппроксимации с использованием ядра Гаусса. Подлежат учету зависимости в виде равенств и неравенств. |