

УДК: 519.8

Принципы устойчивого научного ПО: опыт разработки программы обработки данных малоуглового рассеяния нейтронов

А. Г. Соловьев^a, Т. М. Соловьева^b, А. И. Иваньков^c, А. Х. Исламов^d,
А. И. Куклин^e

Объединенный институт ядерных исследований,
Россия, 141980, г. Дубна, Московская обл., ул. Жолио-Кюри, д. 6

E-mail: ^a solovjev@jinr.ru, ^b tanysha@jinr.ru, ^c ivankov@jinr.ru, ^d akhmed.islamov@gmail.com,
^e alexander.iw.kuklin@gmail.com

Получено 13.02.2026.

Принято к публикации 13.03.2026.

Программа SAS является стандартным инструментом для обработки данных на спектрометре малоуглового рассеяния ЮМО. В статье представлен ретроспективный анализ ее двадцатилетнего развития — от фортрановского прототипа до современной системы. Основное внимание уделено архитектурным решениям, которые обеспечили программе долгосрочную жизнеспособность и возможность адаптации к модернизациям установок.

Ключевым решением стала модульная архитектура, абстрагирующая детекторную систему. Это позволило бесшовно интегрировать данные с двух детекторов, а позднее — и с позиционно-чувствительного детектора. Строгий конвейер преобразований и единое внутреннее представление данных легли в основу физически обоснованных алгоритмов: весового объединения спектров, сглаживания с учетом аппаратного разрешения и встроенного статистического контроля качества. Интерфейсы программы (командная строка для пакетной обработки и графический для интерактивной работы) построены на едином вычислительном ядре, что гарантирует идентичность результатов и гибкость использования.

Длительная эксплуатация подтвердила, что заложенные архитектурные принципы естественным образом соответствуют ключевым характеристикам международного стандарта качества программного обеспечения, включая те, которые критичны для его долгосрочной жизнеспособности. Таким образом, опыт разработки и эволюции SAS демонстрирует универсальный набор архитектурных принципов, которые могут служить основой для создания устойчивого научного программного обеспечения в смежных областях экспериментальной физики.

Ключевые слова: малоугловое рассеяние нейтронов, спектрометр ЮМО, многодетекторная система, конвейерная обработка данных, устойчивое программное обеспечение, объектно-ориентированная архитектура, качество программного обеспечения

UDC: 519.8

Principles of sustainable scientific software: lessons from developing a data processing program for small-angle neutron scattering

A. G. Soloviev^a, T. M. Solovjeva^b, A. I. Ivankov^c, A. Kh. Islamov^d,
A. I. Kuklin^e

Joint Institute for Nuclear Research,
6 Joliot-Curie st., Dubna, Moscow Region, 141980, Russia

E-mail: ^a solovjev@jinr.ru, ^b tanysha@jinr.ru, ^c ivankov@jinr.ru, ^d akhmed.islamov@gmail.com,
^e alexander.iw.kuklin@gmail.com

Received 13.02.2026.

Accepted for publication 13.03.2026.

The SAS program is the primary data processing tool for the YuMO small-angle neutron scattering spectrometer. The paper presents a retrospective analysis of its two-decade evolution, from a Fortran prototype to a modern software system. The analysis focuses on the architectural decisions that have ensured the program's long-term viability and its ability to adapt to instrument upgrades.

The core solution was a modular architecture that abstracts the detector system. This enabled the seamless integration of data from two scattering detectors and, later, from a position-sensitive detector. A strict processing pipeline and a unified internal data representation formed the basis for physically grounded algorithms, including weighted merging of spectra, resolution-aware smoothing, and built-in statistical quality control. The program's interfaces—a command line for batch processing and a graphical user interface for interactive work—are built upon a single computational core, ensuring result consistency and flexibility in use.

Long-term operation has confirmed that the underlying architectural principles naturally align with the key characteristics of international software quality standards, particularly those critical for long-term sustainability. Therefore, the development and evolution of SAS demonstrates a universal set of architectural principles that can serve as a foundation for building sustainable scientific software in related fields of experimental physics.

Keywords: small-angle neutron scattering, YuMO spectrometer, multi-detector system, data processing pipeline, sustainable software, object-oriented architecture, software quality

Citation: *Computer Research and Modeling*, 2026, vol. 18, no. 2, pp. 335–358 (Russian).

Введение

Программа SAS [Soloviev et al., 2026a] является стандартным инструментом для первичной обработки данных на спектрометре малоуглового рассеяния нейтронов (МУРН) ЮМО¹. После модернизации прибора в 2000 году и ввода в строй двух детекторов (с исторически сложившимися названиями OLD и NEW, см. рис. 1) возникла необходимость не просто в новой версии программного обеспечения (ПО), а в принципиально иной архитектуре. Оригинальная фортрановская версия программы², созданная в 1992 году, была жестко завязана на конкретную аппаратную конфигурацию с одним детектором. Это делало ее непригодной для корректного объединения данных двух детекторов без артефактов сшивки и не позволяло адаптироваться к будущим изменениям аппаратуры. Более того, программа была рассчитана на выполнение под операционной системой MS-DOS, что с прекращением ее поддержки создавало критическую угрозу самому существованию ПО для обработки данных на спектрометре.

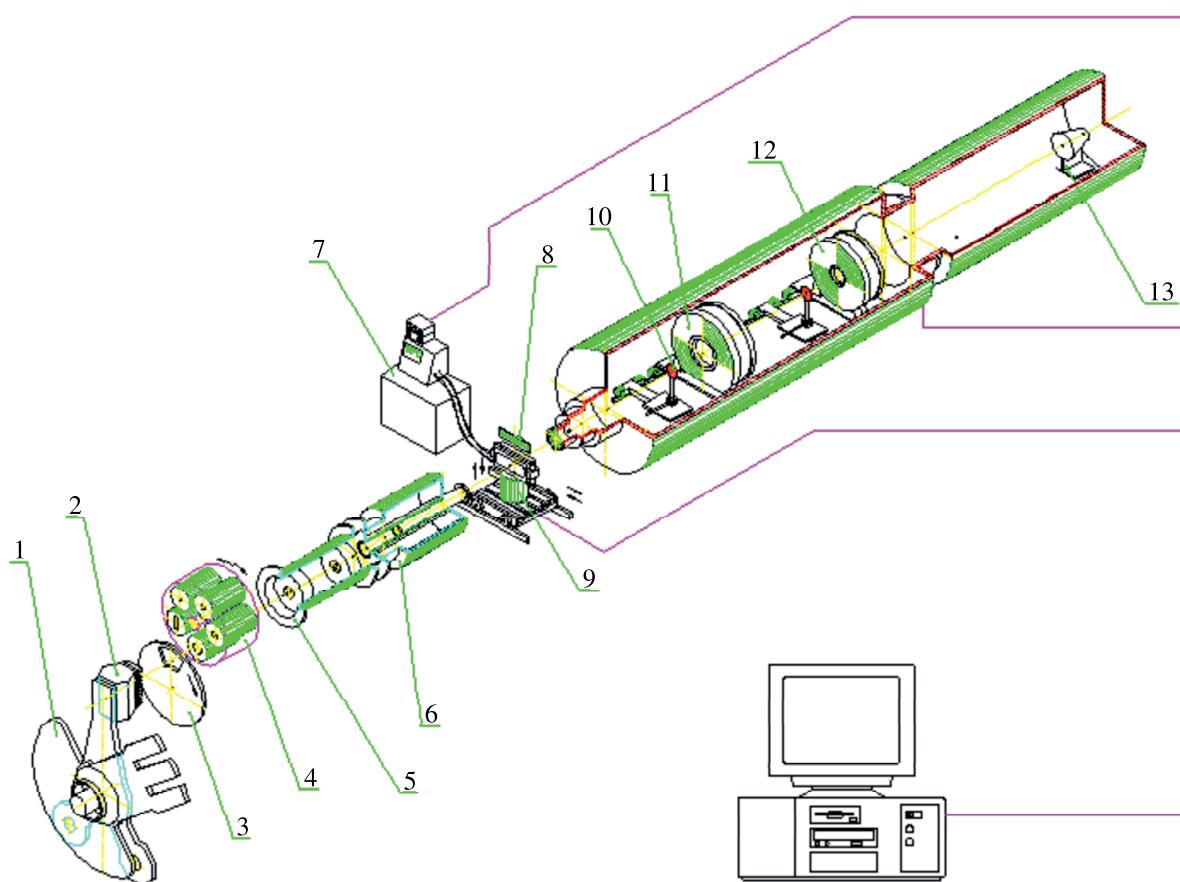


Рис. 1. Схема установки малоуглового рассеяния нейтронов ЮМО: 1 – отражатели, 2 – зона реактора с замедлителем, 3 – прерыватель, 4 – сменный коллиматор, 5 – вакуумная труба, 6 – юстируемый коллиматор, 7 – термостат, 8 – кассета с образцами, 9 – стол образцов, 10 – ванадиевый стандарт, 11, 12 – детекторы (OLD и NEW соответственно), 13 – детектор прямого пучка

Разработка устойчивого (долгоживущего) научного ПО, способного развиваться вместе с экспериментальной установкой, представляет собой сложную инженерную задачу, особенно в условиях быстро меняющейся экспериментальной базы. Большинство специализированных

¹ Установка названа в честь своего создателя Ю. М. Останевича.

² Авторы Ю. М. Останевич и В. Ю. Беззаботнов.

программ для обработки данных физических экспериментов создаются как одноразовые решения, что приводит к их быстрому моральному устареванию при модернизации аппаратуры. В то же время современные стандарты качества ПО (такие как ГОСТ Р ИСО/МЭК 25010 [ГОСТ Р ИСО/МЭК 25010-2015, 2015] и ISO 25010 [ISO/IEC 25010:2011, 2011]) требуют от научного ПО не только корректности вычислений, но и таких характеристик, как сопровождаемость (включая возможность адаптации к новым требованиям) и переносимость. В целом признано, что долгосрочный успех научного программного комплекса зависит от продуманной архитектуры. Однако в литературе недостаточно освещен вопрос о том, какие конкретные архитектурные решения и алгоритмические подходы обеспечивают долгосрочную жизнеспособность специализированного ПО для обработки данных физических экспериментов, особенно в условиях ограниченных ресурсов разработки и поддержки.

Разработка новой версии SAS на C++ началась в 2002 году с осознания этой проблемы: научное ПО должно переживать несколько циклов модернизации экспериментальных установок. Двадцатилетняя история развития SAS представляет собой не просто хронологию исправлений и добавления функций, а последовательную реализацию архитектурных принципов, направленных на создание долгоживущей, сопровождаемой и расширяемой программной платформы. Данная работа представляет собой ретроспективный анализ архитектурных и алгоритмических решений SAS с позиций стандартов качества ПО, что позволяет выделить универсальные принципы разработки устойчивого научного ПО, показать, как они были реализованы в SAS, и продемонстрировать их эффективность на примере двадцатилетней успешной эксплуатации программы.

Физический контекст и функциональность программы SAS

Программа SAS разработана для обработки данных, полученных на малоугловом спектрометре ЮМО, который использует времяпролетный метод (Time-Of-Flight, TOF) [Ostanevich, 1988]. В этом методе длина волны нейтрона λ определяется по времени его пролета t от источника до детектора:

$$\lambda = \frac{ht}{mL},$$

где λ — длина волны (измеряется в ангстремах, \AA), h — постоянная Планка, t — время пролета, m — масса нейтрона, L — пролетная база. Это позволяет одновременно работать с широким спектром длин волн.

Ключевой физической величиной является модуль вектора рассеяния:

$$q = \frac{4\pi \sin\left(\frac{\Theta}{2}\right)}{\lambda} \approx \frac{2\pi\Theta}{\lambda}, \quad (1)$$

где Θ — угол рассеяния, q измеряется в обратных ангстремах (\AA^{-1}).

Дисперсия $\mathbb{D}(q)$ модуля вектора рассеяния, которая возникает из-за конечных геометрических размеров установки и неопределенности во времени пролета нейтронов, вычисляется по формуле, следующей из определения q в (1):

$$\mathbb{D}(q) = q^2 \left[\frac{\mathbb{D}(\Theta)}{\Theta^2} + \frac{1}{12} \left(\frac{\Delta t}{t} \right)^2 \right], \quad (2)$$

где Θ и $\mathbb{D}(\Theta)$ — средний угол рассеяния для соответствующего кольца детектора и его дисперсия, обусловленная конечной шириной кольца, t и Δt — среднее время пролета нейтронов до детектора и ширина временного канала.

Особенностью установки ЮМО является наличие ванадиевого стандарта перед каждым детектором, что позволяет проводить нормировку данных для получения кривой малоуглового рассеяния в абсолютных значениях сечения. Для корректной обработки необходимо учитывать:

- два различных режима работы, применяемых для образцов с существенно различающимся сечением рассеяния (слабо- и сильнорассеивающих);
- влияние мертвых времен детекторов, которое становится значимым при высоких интенсивностях нейтронного пучка;
- аппаратное разрешение $\sqrt{D(q)}$, определяемое геометрией установки;
- необходимость объединения данных от двух детекторов (OLD и NEW) в единый спектр.

За более чем полвека, прошедшего с основополагающей работы Гинье, когда основным получаемым из данных МУРН параметром был радиус инерции (радиус Гинье)¹ [Guinier et al., 1955], цели анализа данных существенно расширились. Появились целые программные комплексы, такие как ATSAS, где результатами становятся определение формы объекта, анализ межчастичных взаимодействий (сгон на нулевую концентрацию) и другие сложные задачи, реализованные командой Д. И. Свергуна и других исследователей [Petoukhov et al., 2007; Petoukhov et al., 2012; Franke et al., 2017; Manalastas-Cantos et al., 2021].

Качество и достоверность таких углубленных анализов существенно зависят от диапазона и точности значений вектора рассеяния q , что в свою очередь выдвигает повышенные требования к первичным данным. Соответственно, программа первичной обработки данных должна всесторонне и с максимальной точностью выполнять фундаментальное преобразование: перевести сырые данные из пространства «счет – время пролета – угол» в физические величины – интенсивность рассеяния в абсолютных единицах (сечение рассеяния $\frac{d\Sigma}{d\Omega}$) и значение вектора рассеяния q . Именно эта точность и воспроизводимость на начальном этапе закладывают основу для всех последующих интерпретаций.

Совокупность описанных требований (как конкретных, связанных с особенностями установки ЮМО, так и общих, обусловленных современными методами анализа) формирует ключевые задачи для программной реализации: обеспечение различных схем нормировки, введение поправок на неидеальность аппаратуры, учет физики измерений в преобразованиях и бесшовное слияние разнородных данных.

Программа SAS обеспечивает полный цикл первичной обработки данных: от чтения сырых спектров до получения сечения рассеяния в абсолютных единицах в пространстве q . Ее функциональность включает коррекцию на мертвые времена, вычитание фона, нормировку на ванадиевый стандарт, преобразование в пространство q и усреднение данных. Программа поддерживает различные режимы нормировки и позволяет исключать из анализа поврежденные кольца детекторов.

Эти физические требования и функциональные задачи непосредственно определили архитектурные решения, которые рассматриваются в следующем разделе.

Архитектура как основа устойчивого ПО

Разработка началась с реверс-инжиниринга оригинальной фортрановской программы, но быстро вышла за рамки простого портирования. Основной задачей стало создание такой архитектуры, которая позволила бы программе развиваться вместе со спектрометром, абстрагируясь от конкретных деталей аппаратуры и форматов данных.

¹ Радиус инерции (радиус Гинье) — ключевой параметр, характеризующий усредненный размер рассеивающих частиц в растворе. Определяется из анализа начального участка кривой рассеяния в представлении Гинье.

В отличие от многих специализированных программ для обработки данных физических экспериментов, создававшихся как монолитные решения под конкретную аппаратную конфигурацию, SAS изначально проектировалась с учетом необходимости адаптации к будущим изменениям. Ключевым решением стало выделение класса `RingDetector`, инкапсулирующего все аппаратно зависимые параметры, и построение на его основе строгого конвейера обработки данных. Такой подход, реализованный в 2002–2003 годах [Соловьев и др., 2003], оказался созвучен современным принципам модульной архитектуры и четкого разделения ответственности, которые впоследствии получили развитие в стандартах качества ПО и практике разработки устойчивого научного ПО, от ранних примеров [Blatt et al., 2013] до современных исследований [Eisty et al., 2025; Zech, Pekaric, 2025].

Объектно-ориентированная модель детекторной системы

Все аппаратные константы спектрометра были вынесены в единый класс `RingDetector`, представляющий отдельное кольцо детектора. Каждый детектор (OLD, NEW, а также детектор прямого пучка¹) реализуется как коллекция объектов-колец, содержащих свои геометрические параметры, мертвые времена, расстояния и калибровочные коэффициенты. Общие для спектрометра в целом параметры хранятся в конфигурации программы (`QSettings`).

Ключевым усовершенствованием стал переход от раздельной обработки данных с детекторов OLD и NEW к единой системе, где элементарным объектом выступает кольцо детектора [Soloviev et al., 2017]. Это позволило избежать проблемы сшивки спектров, поскольку программа обрабатывает все кольца одновременно, а не объединяет данные постфактум.

Иерархия классов, непосредственно реализующих обработку спектров, наследуется от `RingDetector`. Такой подход обеспечил возможность поддержки различных типов детекторов в рамках единой вычислительной модели и позволил легко добавлять новые детекторы без изменения ядра обработки данных, поскольку все необходимые параметры доступны в объекте кольца на любом этапе обработки. Это архитектурное решение также заложило основу для естественной поддержки позиционно-чувствительного детектора (ПЧД), реализованной через конвертер данных PSD2SAS (Position-Sensitive Detector to SAS) [Soloviev et al., 2018].

Конвейерная архитектура обработки данных

Оригинальный алгоритм обработки был разбит на последовательные, логически завершённые этапы. В новой версии программы эти этапы реализованы как иерархия классов, наследующихся от `RingDetector`:

- `RawRingSpectrum` — исходные данные кольца детектора, считанные из файла;
- `CorRingSpectrum` — скорректированные данные (поправка на мертвые времена, нормировка на ширину каналов, вычитание фона);
- `NorRingSpectrum` — данные, нормированные на ванадиевый стандарт данные с поддержкой нескольких режимов нормировки;
- `QRingSpectrum` — данные, преобразованные в пространство модуля вектора рассеяния q ;
- `AveSpectrum` и `SrtSpectrum` — финальные спектры после объединения данных со всех колец: усредненные или отсортированные по q , готовые для физического анализа.

¹ Детектор прямого пучка, в отличие от детекторов рассеяния, используется не для регистрации рассеянных нейтронов, а для измерения пропускания образца, используемого, в частности, в процедуре нормировки сильнорассеивающих образцов.

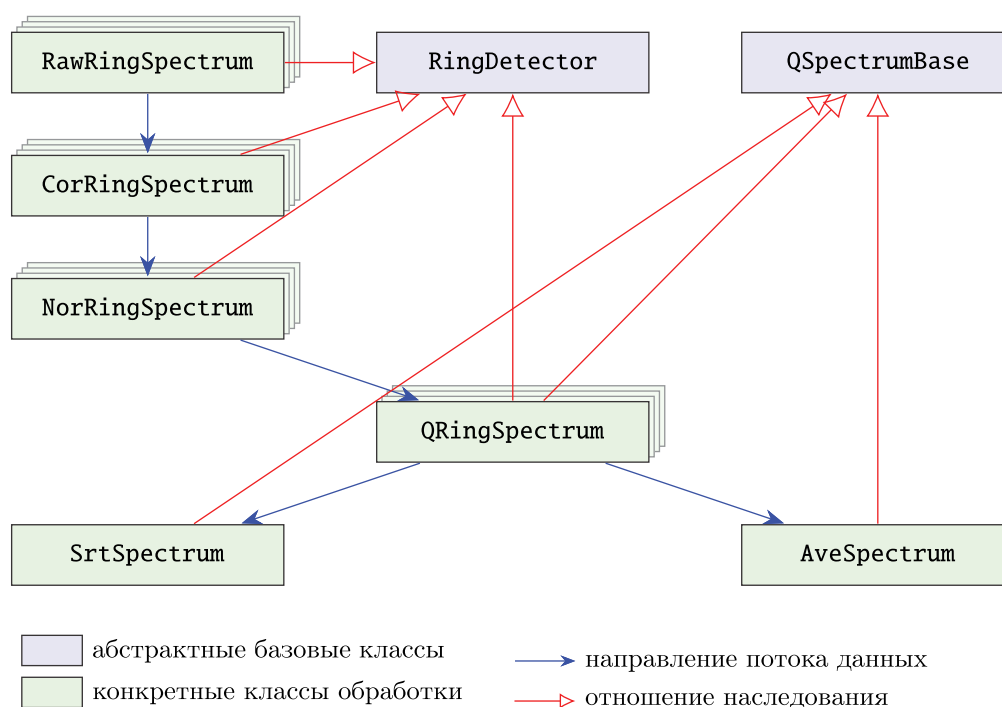


Рис. 2. Конвейер обработки данных в SAS

Архитектура обработки данных образует строгий конвейер преобразований, где каждый следующий класс конструируется на основе предыдущего (см. рис. 2). Каждый класс отвечает за строго определенный этап обработки. Такая архитектура гарантирует:

- **строгую последовательность преобразований:** невозможно пропустить критический этап обработки;
- **изоляция ответственности:** изменение одного этапа не требует модификации других;
- **повторное использование:** общие операции вынесены в базовые классы;
- **тестируемость:** каждый класс может быть протестирован независимо.

Класс `NorRingSpectrum` поддерживает несколько конструкторов для различных режимов нормировки (стандартная, для сильно рассеивающих образцов, в том числе с учетом пропуска образца), что демонстрирует гибкость архитектуры. Класс `QSpectrumBase` предоставляет единый интерфейс для работы с данными в пространстве q , включая операцию сглаживания с учетом аппаратного разрешения.

Единая архитектура внутреннего представления данных

Для эффективной и согласованной работы с экспериментальными данными на всех этапах конвейера была разработана библиотека шаблонов для работы с одномерными гистограммами HTL (Histogram Template Library) [Soloviev, 2026b]. Стремление к легковесности, минимальной зависимости от внешних компонентов и гарантированной кросс-платформенности определило выбор в пользу собственной реализации вместо использования фреймворка ROOT, который на старте разработки SAS в 2002 году не отвечал этим требованиям в полной мере. Библиотека HTL предоставляет унифицированные классы для хранения и обработки одномерных распределений (гистограмм), что стало основой для обработки данных от различных поколений аппаратуры в рамках единой вычислительной модели.

Ядро библиотеки составляют три шаблонных класса, разделяющих ответственность.

- `htl::domain<Tx>` — задает область определения гистограммы (диапазон, размеры бинов) и обеспечивает операции над ней: линейные преобразования, быстрый поиск и изменение структуры (ребиннинг).
- `htl::histogram<Tx, Ty>` — реализует гистограмму, объединяя область определения (`domain`) и контейнер данных, и обеспечивает безопасный доступ с проверкой границ и функциональный доступ по значению аргумента.
- `htl::bin<T>` — хранит пару (значение интенсивности и его дисперсию), представляющую один элемент гистограммы. Класс перегружает арифметические операции для корректного учета погрешностей.

Такое разделение позволяет независимо изменять логику хранения (`histogram`), математику погрешностей (`bin`) и структуру разбиения (`domain`). Использование шаблонов обеспечило типобезопасность и возможность оптимизации на этапе компиляции.

Архитектура HTL, основанная на динамических контейнерах стандартной библиотеки C++ (STL), позволила SAS работать с данными на произвольной временной шкале. Это было недоступно в оригинальной версии, которая поддерживала только 256 каналов, где ширина канала удваивалась через каждые 64 канала. Использование HTL позволяет единообразно обрабатывать данные, содержащие произвольное количество временных каналов и произвольную ширину этих каналов:

- старые данные: 256 каналов с удвоением ширины (обратная совместимость);
- современные данные: 1024 канала;
- данные с ПЧД: 512 каналов при текущих настройках системы, но конфигурация может меняться.

Класс `domain` позволяет гибко управлять группировкой бинов (ребиннинг). Это используется, например, на этапе нормировки для увеличения статистической значимости данных в областях с низким сигналом.

Библиотека HTL стала фундаментальным строительным блоком системы, обеспечивающим целостность данных на всем пути конвейера:

- классы временных спектров (`RawRingSpectrum`, `CorRingSpectrum`, `NorRingSpectrum`) используют специализацию `htl::histogram<double, htl::bin<double>` > для хранения значений интенсивности и их статистических погрешностей;
- классы, работающие в пространстве q (производные от `QSpectrumBase`), используют массивы объектов `htl::bin<double>`, что автоматически обеспечивает корректный учет погрешностей во всех арифметических операциях;
- все базовые математические и статистические операции (вычисление моментов, интегралов и т. п.) реализованы на базе HTL, обеспечивая единый математический аппарат для работы со спектрами.

Таким образом, механизм автоматического учета погрешностей, инкапсулированный в `htl::bin`, создает необходимую математическую основу для корректной реализации алгоритмов физической обработки, таких как весовое объединение данных от разных детекторов или встроенный контроль качества (χ^2 -критерий), описанных далее.

Форматы данных и обратная совместимость

Эволюция внешних форматов хранения данных стала естественным следствием ключевых архитектурных решений SAS. Переход к объектной модели, где элементарной единицей является кольцо детектора (RingDetector), в сочетании с использованием библиотеки HTL для единого внутреннего представления, позволил переосмыслить организацию данных. Оригинальная фортрановская версия работала с двумя отдельными бинарными файлами (параметры + данные) для детекторов OLD или NEW по отдельности. Современная версия принимает два файла: XML-файл, содержащий параметры измерений, и соответствующий бинарный файл с данными. Такой подход заложил основу для естественной поддержки ПЧД, данные которого могут быть преобразованы в такой формат с помощью конвертера PSD2SAS [Soloviev et al., 2018]. Кроме того, использование XML для описания параметров обеспечивает движение в сторону открытых форматов. Это делает параметры эксперимента доступными для чтения и проверки вне одной программы. При этом SAS сохраняет обратную совместимость, поддерживая оба формата входных данных.

Алгоритмы с учетом физики установки

Описанная модульная архитектура не только обеспечила гибкость и сопровождаемость, но и создала основу для реализации алгоритмов, глубоко интегрирующих физическую модель установки в вычислительные процедуры. Алгоритмы SAS были разработаны с учетом конкретных характеристик спектрометра ЮМО: его геометрии, разрешения, особенностей детекторной системы и статистической природы нейтронных измерений.

Встроенный контроль качества данных

Уже в 2003 году в алгоритмы был интегрирован расчет χ^2 как критерий согласованности спектров [Соловьев и др., 2003]. Эта функция является ключевым элементом системы валидации, обеспечивающим контроль качества на самом раннем этапе. Основная практическая необходимость такой проверки связана с тем, что измерения в эксперименте проводятся сериями, и для получения итоговой статистики данные от разных сеансов должны суммироваться. Некорректное сложение спектров, существенно различающихся из-за аппаратных сбоев, изменений условий эксперимента или проблем с детекторами, привело бы к систематическим ошибкам в окончательных результатах. Поэтому автоматическое выявление и отсев аномальных данных до их объединения являются критически важной задачей.

Критерий χ^2 для количественного сравнения двух нормированных спектров в заданном диапазоне каналов вычисляется по формуле

$$\chi^2 = \sum_{j=N_{\min}}^{N_{\max}} \frac{\left(\frac{I_1[j]}{N_1} - \frac{I_2[j]}{N_2} \right)^2}{\mathbb{D} \left(\frac{I_1[j]}{N_1} - \frac{I_2[j]}{N_2} \right)},$$

где $I_1[j]$, $I_2[j]$ — интенсивности в канале j для первого и второго спектров соответственно; $\mathbb{D}(x)$ — дисперсия величины x , которая хранится и рассчитывается в рамках объекта `htl::bin<double>`; N_1 и N_2 — нормировочные множители, обеспечивающие сравнение формы спектров, а не их абсолютной светимости:

$$N_1 = \sum_{j=N_{\min}}^{N_{\max}} I_1[j], \quad N_2 = \sum_{j=N_{\min}}^{N_{\max}} I_2[j];$$

N_{\min} , N_{\max} — границы анализируемого диапазона каналов.

Критерий является классическим χ^2 -тестом, где нормировка на N_1 и N_2 позволяет сравнивать формы спектров независимо от их общей интенсивности. Это особенно важно при обработке данных от разных сеансов измерений, которые могут иметь различную статистическую значимость.

Для принятия решения о совместимости спектров вычисляется нормированное отклонение:

$$d = \frac{|\chi^2 - M|}{\sqrt{2M}},$$

где $M = N_{\max} - N_{\min}$ — число степеней свободы. Статистическое обоснование этого критерия основано на свойствах распределения χ^2 : для M степеней свободы ожидаемое значение — $\mathbb{E}[\chi^2] = M$, а дисперсия — $\mathbb{D}[\chi^2] = 2M$. При достаточно больших M распределение χ^2 аппроксимируется нормальным, и величина d представляет собой нормированное отклонение наблюдаемого χ^2 от ожидаемого значения.

Спектры считаются статистически совместимыми и пригодными для последующего совместного анализа, если выполняется условие $d < d_0$. Пороговое значение d_0 задается в конфигурационном файле системы, что обеспечивает гибкость; значение по умолчанию равно 6. Это соответствует отклонению в 6σ в приближении нормального распределения, что означает вероятность случайного получения такого отклонения менее $3 \cdot 10^{-9}$. Таким образом, по умолчанию отбраковываются только спектры с заведомо неслучайными различиями, обусловленными аппаратными сбоями или изменением условий эксперимента.

Вычисление нормированного отклонения d программно реализовано в виде функции `distance()`, предназначенной для попарного сравнения спектров, что обеспечивает его легкую интеграцию в конвейер обработки. Это позволяет автоматически проверять согласованность исходных данных от разных сеансов измерений и отбраковывать заведомо некорректные наборы данных еще до этапов усреднения или сложения. Таким образом, статистический критерий превращается в активный инструмент оперативного контроля качества, встроенный в ядро системы анализа.

Кроме статистического контроля согласованности спектров важным инструментом управления качеством данных является возможность избирательного отключения отдельных колец детектора. Эта функция, доступная через командную строку (опция `-without-ring`) или графический интерфейс, решает другую практическую задачу: она позволяет исключать из анализа данные от поврежденных, некалиброванных или нестабильно работающих элементов детекторной системы. Такое исключение выполняется без необходимости модификации исходных файлов или повторения измерений, что существенно повышает гибкость и отказоустойчивость процесса обработки.

Нормировка на ванадиевый стандарт

В SAS реализованы два основных алгоритма нормировки данных на ванадиевый стандарт, соответствующие разным типам образцов.

- **Стандартная процедура для слаборассеивающих образцов**

Для образцов с сечением рассеяния менее 10 см^{-1} используется процедура, основанная на чередующихся измерениях с ванадиевым стандартом и без него [Куклин и др., 2006].

Интенсивности на детекторе определяются как

$$I_s = I_0(\lambda)\epsilon(\lambda)T_s \left(\frac{d\Sigma}{d\Omega} \right)_s \Omega_s d_s,$$

$$I_{s+v} = I_s + I_0(\lambda)\epsilon(\lambda)T_v T_s \left(\frac{d\Sigma}{d\Omega} \right)_v \Omega_v d_v,$$

где $I_0(\lambda)$ — интенсивность нейтронного пучка на образце, $\epsilon(\lambda)$ — эффективность детектора, T_s, T_v — пропускания образца и ванадия, d_s, d_v — толщины образца и ванадия, Ω_s, Ω_v — телесные углы.

Нормированное сечение рассеяния вычисляется по формуле

$$\frac{\left(\frac{d\Sigma}{d\Omega} \right)_s}{\left(\frac{d\Sigma}{d\Omega} \right)_v} = \frac{I_s}{I_{s+v} - I_s} \frac{T_v \Omega_v d_v}{\Omega_s d_s}.$$

• Процедуры для сильнорассеивающих образцов

Для образцов с большим сечением рассеяния используется одна из следующих процедур [Куклин и др., 2006].

- **Нормировка на детектор прямого пучка.** Используется дополнительное измерение «пустого пучка» (только ванадиевый стандарт). В этом случае

$$\frac{\left(\frac{d\Sigma}{d\Omega} \right)_s}{\left(\frac{d\Sigma}{d\Omega} \right)_v} = \frac{I_s T_s d_v \Omega_v}{I_v T_v d_s \Omega_s},$$

где I_v — интенсивность от ванадиевого стандарта без образца. Пропускание образца T_s определяется по данным детектора прямого пучка:

$$T_s = \frac{I_s^d}{I_e^d},$$

где I_s^d и I_e^d — интенсивности на детекторе прямого пучка с образцом и без него соответственно.

- **Нормировка на ванадиевый рассеиватель.** В данном случае используется процедура, основанная на чередующихся измерениях растворителя (буфера) с ванадиевым стандартом и без него. Интенсивности на детекторе определяются как

$$I_s = I_0(\lambda)\epsilon(\lambda)T_s \left(\frac{d\Sigma}{d\Omega} \right)_s \Omega_s d_s,$$

$$I_b = I_0(\lambda)\epsilon(\lambda)T_b \left(\frac{d\Sigma}{d\Omega} \right)_b \Omega_b d_b,$$

$$I_{b+v} = I_b + I_0(\lambda)\epsilon(\lambda)T_v T_b \left(\frac{d\Sigma}{d\Omega} \right)_v \Omega_v d_v,$$

где $I_0(\lambda)$ — интенсивность нейтронного пучка на образце, $\epsilon(\lambda)$ — эффективность детектора, T_s, T_b, T_v — пропускания образца, буфера и ванадия, d_s, d_v, d_b — толщины образца, буфера и ванадия, Ω_s, Ω_v — телесные углы.

Нормированное сечение рассеяния вычисляется по формуле

$$\frac{\left(\frac{d\Sigma}{d\Omega} \right)_s}{\left(\frac{d\Sigma}{d\Omega} \right)_v} = \frac{I_s}{I_{b+v} - I_b} \frac{T_v T_b \Omega_v d_v}{T_s \Omega_s d_s}.$$

Алгоритмы нормировки реализованы в классе `NorRingSpectrum` и основаны на делении гистограмм с данными образца на гистограммы с данными ванадиевого стандарта.

Методы повышения качества временных спектров

Поскольку операция деления гистограмм чувствительна к статистическим флуктуациям, особенно в области малых интенсивностей, в SAS предусмотрены специальные методы повышения качества данных, применяемые как перед нормировкой, так и после нее.

- **Объединение временных каналов.** Для увеличения статистической значимости данных перед нормировкой применяется адаптивное объединение соседних временных каналов. Эта процедура позволяет повысить статистическую значимость в областях с низким сигналом (большие времена пролета, малые q) за счет суммирования интенсивности, что достигается ценой снижения временного разрешения. Реализация основана на операциях с доменами библиотеки HTL и предоставляет два режима.
 - **Удвоение ширины каналов:** воспроизводится жестко заданная временная шкала оригинальной фортрановской программы. Исходная область определения с равномерными 1024 каналами преобразуется в область, в которой ширина канала ступенчато удваивается каждые 64 канала от начала к концу (результатирующее число каналов получается близким к 256 каналам фортрановской версии).
 - **Пользовательское задание групп каналов:** экспериментатор может задавать произвольные схемы объединения. Реализация использует словарь, где ключами являются границы групп каналов, а значениями — кратности объединения в каждой группе.

Применение измененного домена к гистограмме через метод `rebinned()` автоматически пересчитывает данные с учетом новой структуры бинов, суммируя интенсивности в объединяемых каналах. Такой подход, основанный на операциях с доменами HTL, обеспечивает типобезопасность, автоматический учет погрешностей и возможность повторного использования для различных типов спектров. Пользователь может выбирать между предопределенной адаптивной схемой и полностью настраиваемым объединением, что позволяет оптимально балансировать между статистической точностью и временным разрешением для конкретного эксперимента.

- **Медианное сглаживание временных спектров.** Для подавления случайных выбросов применяется скользящее медианное сглаживание:
 - в классе `CorRingSpectrum` — сглаживание скорректированных временных спектров перед нормировкой, что устраняет артефакты, связанные с аппаратными сбоями;
 - в классе `NorRingSpectrum` — сглаживание уже нормированных временных спектров для устранения артефактов, возникающих при делении на малые значения.

Применение этих методов помогает улучшить качество финальных спектров в пространстве q :

- снижается уровень случайных флуктуаций в области больших времен пролета (малых q), где статистика наименьшая;
- сокращается число каналов, отбрасываемых из-за нулевого счета ванадия;
- повышается воспроизводимость результатов при обработке повторных измерений;
- улучшается соотношение «сигнал/шум» в финальных данных без потери физически значимой информации.

Весовой алгоритм объединения данных от разных детекторов

С вводом двух детекторов (OLD и NEW) [Kuklin et al., 2005] возникла проблема математически корректного объединения их данных. Простое склеивание спектров приводило к артефактам на стыке диапазонов.

В оригинальной программе SAS уже существовал весовой алгоритм, разработанный для объединения колец одного детектора. Этот алгоритм был восстановлен по фортрановской реализации и обобщен для совместной обработки данных обоих детекторов в рамках единой архитектуры, представленной в предыдущем разделе. Основная идея алгоритма заключается во взвешенном усреднении данных в области q , где перекрываются измерения от разных колец. Веса определяются обратно пропорционально квадрату статистической погрешности.

Предварительно для каждого кольца производится переход в пространство q по формуле (1), где в качестве Θ используется средний угол рассеяния для данного кольца. Далее экспериментатор задает интервал $[q_{\min}, q_{\max}]$ и желаемое разбиение по q (равномерное, логарифмическое или квадратичное). Пусть отрезки этого разбиения имеют вид $[q_n, q_{n+1})$, где $n = 0, 1, \dots, N-1$, а N — заданное количество точек в усредненном спектре. В каждый такой отрезок может попасть несколько точек от разных колец. Для каждого отрезка вычисляется усредненное значение по формулам

$$q[n] = \frac{1}{W_n} \sum_i \sum_{j=N_{\text{begin}}}^{N_{\text{end}}} \frac{q_i[j]}{\mathbb{D}(I_i[j])},$$

$$I[n] = \frac{1}{W_n} \sum_i \sum_{j=N_{\text{begin}}}^{N_{\text{end}}} \frac{I_i[j]}{\mathbb{D}(I_i[j])},$$

$i, j: q_i[j] \in [q_n, q_{n+1})$

где

$$W_n = \sum_i \sum_{j=N_{\text{begin}}}^{N_{\text{end}}} \frac{1}{\mathbb{D}(I_i[j])},$$

$i, j: q_i[j] \in [q_n, q_{n+1})$

i — индекс кольца детектора, j — номер канала в спектре этого кольца в пространстве q .

Дисперсии для усредненных величин и вычисляются как

$$\mathbb{D}(I[n]) = \frac{1}{W_n},$$

$$\mathbb{D}(q[n]) = \frac{1}{W_n} \sum_i \sum_{j=N_{\text{begin}}}^{N_{\text{end}}} \frac{\mathbb{D}(q_i[j])}{\mathbb{D}(I_i[j])}.$$

$i, j: q_i[j] \in [q_n, q_{n+1})$

Суммирование производится только для ненулевых значений $q_i[j]$, $I_i[j]$ и $\mathbb{D}(I_i[j])$. Используемые дисперсии во временных спектрах корректно обрабатываются через класс `htl::bin<double>`.

Для неперекрывающихся областей q данные используются без изменений. В области перекрытия алгоритм обеспечивает плавный переход, где вклад каждой точки определяется ее статистической значимостью. Алгоритм поддерживает линейную, логарифмическую и квадратичную шкалы по Q , а также режим без усреднения (сортировка точек по q).

В результате достигается плавное объединение данных в расширенном диапазоне q без потери разрешения и артефактов сшивки, что существенно повышает точность последующего анализа.

Сглаживание данных с учетом аппаратного разрешения

Для подавления статистического шума без потери физически значимой информации в SAS реализована процедура адаптивного сглаживания. Метод, предложенный в работе [Soloviev et al., 2003], основывается на формализме непрерывного вейвлет-преобразования (Continuous Wavelet Transform, CWT) с гауссовым ядром, используемым вместо классического вейвлета. Ключевая особенность реализации в SAS — параметризация ширины этого ядра в соответствии с локальным аппаратным разрешением спектрометра, что принципиально отличает метод от стандартных фильтров.

Аппаратное разрешение спектрометра характеризуется дисперсией $\mathbb{D}(q)$ модуля вектора рассеяния, вычисляемой по формуле (2). Физическая обоснованность метода заключается в том, что ширина сглаживающего ядра a задается пропорционально среднеквадратичному отклонению:

$$a = \alpha \cdot \sqrt{\mathbb{D}(q)},$$

где безразмерный коэффициент α определяет степень сглаживания (по умолчанию $\alpha = 1$, что соответствует подавлению шума в пределах аппаратного разрешения). Таким образом, ширина ядра автоматически адаптируется к локальной разрешающей способности установки в каждой точке q .

Алгоритмически процедура выполняется после преобразования данных в пространство q :

- 1) для каждой точки $q[i]$ определяется локальный масштаб $a_i = \alpha \cdot \sqrt{\mathbb{D}(q[i])}$;
- 2) сглаженное значение интенсивности в этой точке вычисляется как взвешенное среднее с гауссовыми весами:

$$I_{\text{smooth}}[i] = \frac{\sum_j I[j] \cdot \exp\left[-\frac{1}{2} \left(\frac{q[j]-q[i]}{a_i}\right)^2\right]}{\sum_j \exp\left[-\frac{1}{2} \left(\frac{q[j]-q[i]}{a_i}\right)^2\right]},$$

где суммирование ведется по всем точкам j того же спектра;

- 3) статистические погрешности исходных данных трансформируются с учетом соответствующих весов.

На рис. 3 интерфейс программы визуализирует результат применения этой процедуры. Ключевые преимущества данного подхода заключаются в следующем.

- **Адаптивность:** степень сглаживания определяется характеристиками установки, а не подбирается эмпирически.
- **Сохранение информации:** исключается излишнее подавление сигнала, которое могло бы исказить физически значимые особенности спектра.
- **Улучшение точности моделирования:** как показано в оригинальной работе [Soloviev et al., 2003], использование такого подхода снижает значение χ^2 при фитировании модельных кривых на 20–40 % по сравнению с обработкой нефитированных данных.

Таким образом, параметризация сглаживающего ядра через аппаратное разрешение $\sqrt{\mathbb{D}(q)}$ обеспечивает органичное встраивание физической модели в вычислительный конвейер SAS.

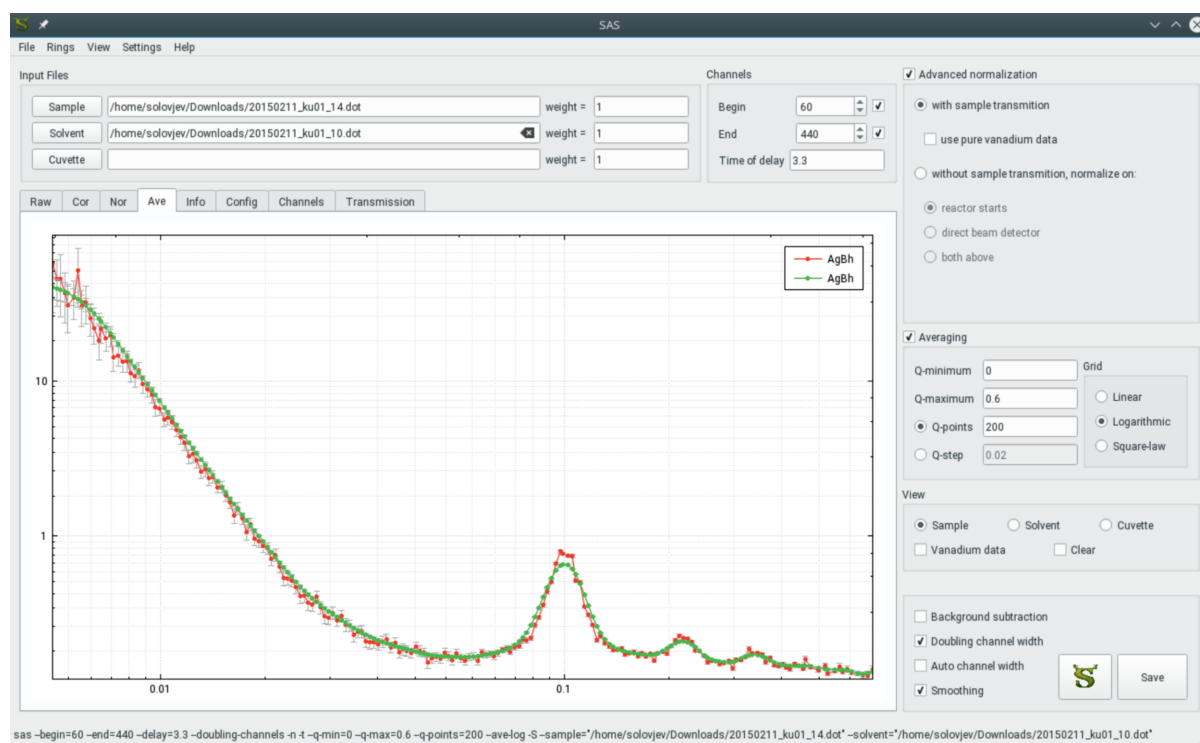


Рис. 3. Программа SAS для обработки данных ЮМО. Можно видеть также результаты объединения данных с разных детекторов (красная линия) и сглаживания (зеленая линия), описанные в предыдущих разделах

Эволюция пользовательского интерфейса

Эволюция интерфейсов SAS основана на ключевом архитектурном решении: единое вычислительное ядро, вызываемое из различных интерфейсных оболочек. Фактически SAS представляет собой единую программу, которая в зависимости от аргументов командной строки запускается либо в режиме командной строки (CLI), либо в режиме графического интерфейса (GUI). Это гарантирует идентичность результатов независимо от способа взаимодействия с пользователем и создает основу для независимого развития интерфейса.

Интерфейс командной строки и пакетная обработка

Технической основой этой архитектуры является интерфейс командной строки (CLI), который предоставляет полный программный контроль над обработкой. Его реализация эволюционировала от библиотеки `getopt` в ранних версиях до `QCommandLineParser` после перехода на Qt, сохраняя строгую валидацию аргументов и единообразную справку (`-help`). CLI служит не просто одним из интерфейсов, а базовым механизмом вызова ядра. Именно в виде командной строки ядро принимает все параметры обработки. Поэтому графический (Qt) и веб-интерфейсы реализованы как надстройки, чья задача — сформировать корректную командную строку на основе действий пользователя и выполнить ее, запустив ту же самую программу. Такой подход делает CLI естественным инструментом для автоматизации и пакетной обработки, а GUI — удобным решением для интерактивной работы, при этом оба используют одно и то же вычислительное ядро.

Развитие графического интерфейса

Развитие графического интерфейса SAS отражает поиск оптимального технического решения для согласования функциональности CLI и GUI — задачи, возникающей при добавлении

каждой новой опции в ядро программы. Первая графическая оболочка, созданная уже в 2003 году, была отдельной программой для Windows, формировавшей и выполнявшей командную строку SAS. Поддержка такой реализации фактически означала сопровождение двух независимых программ, что усложняло разработку и не исключало риск их рассогласования.

В 2008 году появилась кроссплатформенная версия GUI на основе библиотеки FLTK [FLTK]. Визуализация в этой версии осуществлялась библиотекой GLUPLOT [Soloviev, 2026a] на основе OpenGL [OpenGL]. Она также оставалась отдельной программой, но была интегрирована в единый механизм сборки с основной программой, что упростило процесс разработки и поддержки.

Логическим завершением этой эволюции стал переход на Qt [Qt] в 2014 году, который позволил не только создать современный интерфейс (рис. 3), но и реализовать окончательный вариант архитектуры — единую программу, где CLI и GUI стали режимами одного исполняемого файла.

Прототип веб-приложения

Как одно из возможных направлений эволюции интерфейса в 2025 году был создан прототип веб-приложения SAS (рис. 4), демонстрирующий возможность удаленного доступа к обработке данных через браузер.

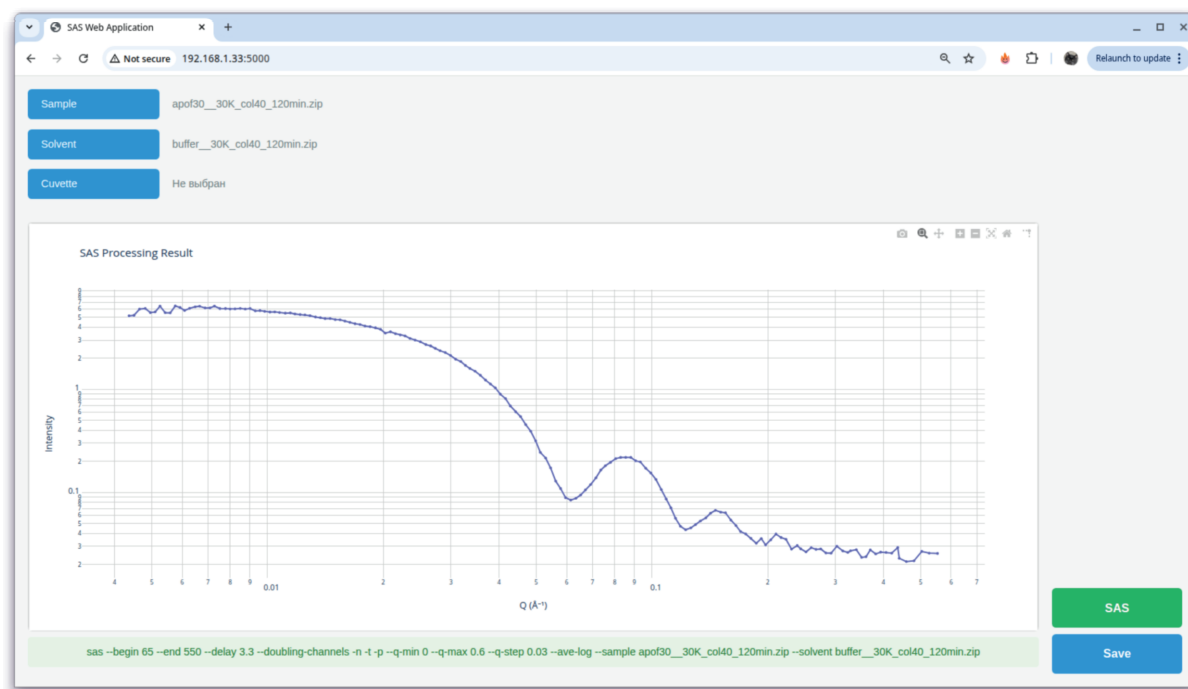


Рис. 4. Прототип веб-интерфейса SAS

Архитектура прототипа построена по классической трехзвенной схеме:

- 1) **клиентская часть** (HTML/CSS/JavaScript) — минималистичный интерфейс на чистом JavaScript с использованием библиотеки Plotly для интерактивной визуализации;
- 2) **серверная часть** (Python/Flask) — приложение, принимающее файлы данных и формирующее командную строку для вызова SAS;

- 3) **вычислительное ядро** — нативная программа SAS, вызываемая через API модуля subprocess.

Ключевая особенность этой схемы — сохранение единого вычислительного ядра: веб-интерфейс лишь формирует параметры командной строки и вызывает тот же исполняемый файл SAS, что используется в десктопной версии. Это гарантирует полную идентичность результатов независимо от способа доступа. Техническая реализация включает асинхронную загрузку файлов через AJAX, валидацию входных данных на стороне сервера и автоматическое скачивание результатов.

Таким образом, прототип веб-приложения служит практическим подтверждением возможности легкого расширения SAS на принципиально новые типы интерфейсов. Полноценное развертывание подобных сервисов потребует дополнительного анализа и реализации мер безопасности, что выходит за рамки данной работы.

Архитектурные решения SAS в свете стандартов качества ПО

Анализ архитектуры SAS в контексте современных стандартов качества ПО показывает, что решения, принятые в начале разработки, соответствуют принципам, которые позднее были систематизированы в стандартах. Архитектура SAS, сформировавшаяся в первые годы разработки, естественным образом реализует ключевые характеристики национального стандарта ГОСТ Р ИСО/МЭК 25010 [ГОСТ Р ИСО/МЭК 25010-2015, 2015] и его международного аналога ISO 25010 [ISO/IEC 25010:2011, 2011].

Функциональная пригодность

Программа SAS удовлетворяет потребности пользователей за счет реализации специфических алгоритмов обработки данных малоуглового рассеяния. Соответствие стандарту проявляется в следующих аспектах.

- **Функциональная полнота:** система предоставляет полный конвейер обработки — от загрузки сырых данных и их валидации до получения спектра $I(q)$ (зависимость интенсивности от модуля волнового вектора), готового для анализа.
- **Функциональная корректность:** строгая математическая реализация преобразования сырых данных в физические величины и весового объединения спектров исключает систематические сдвиги и артефакты сшивки.
- **Функциональная целесообразность:** реализованные функции напрямую соответствуют задачам эксперимента на спектрометре ЮМО, включая нормировку на ванадиевый стандарт, весовое объединение данных двух (и более) детекторов и учет аппаратного разрешения при сглаживании.

Уровень производительности

Эффективность использования ресурсов обеспечивается на архитектурном и алгоритмическом уровнях.

- **Временные характеристики:** обработка одного измерения занимает менее 1 секунды на современном оборудовании. Это достигается за счет алгоритмической оптимизации вычислительно интенсивных операций, дополнительные низкоуровневые оптимизации не требуются. Поддержка пакетной обработки через интерфейс командной строки обеспечивает обработку серии измерений без участия пользователя, что полностью удовлетворяет потребностям эксперимента.

- **Использование ресурсов:** потребление памяти соответствует объему входных данных (менее 1 МБ на измерение), поскольку архитектура минимизирует копирование между этапами конвейера и использует эффективное внутреннее представление через библиотеку HTL.

Подхарактеристика «**потенциальные возможности**» не оценивалась, поскольку требования к предельным значениям параметров не предъявлялись и не влияют на целевое применение программы в условиях реального эксперимента.

Совместимость

Способность SAS взаимодействовать с другими компонентами ПО эксперимента реализуется через **функциональную совместимость** на уровне данных.

- Программа принимает данные в современном формате (XML-файл параметров + бинарный файл данных), сохраняя обратную совместимость с устаревшим форматом (два бинарных файла для каждого детектора). Данные от ПЧД принимаются через конвертер PSD2SAS [Soloviev et al., 2018].
- Результаты обработки сохраняются в простом текстовом формате, что обеспечивает их использование другими программами анализа, такими как Fitter [Soloviev et al., 2026b].
- Программа может принимать данные напрямую из программного комплекса управления спектрометром Sonix+ [Кирилов и др., 2004; Кирилов и др., 2024]. Благодаря интерфейсу командной строки она интегрируется в этот комплекс, что позволяет получать предварительные спектры для оперативного управления экспериментом.

Подхарактеристика «**сосуществование**» не рассматривается, поскольку в условиях эксплуатации отсутствуют конфликты при совместном использовании ресурсов с иным программным обеспечением.

Удобство использования

Программа SAS обеспечивает удобство работы благодаря реализации следующих подхарактеристик качества.

- **Определимость пригодности:** графический интерфейс визуализирует данные на всех этапах обработки, позволяя пользователю немедленно оценить корректность промежуточных результатов и соответствие программы решаемым задачам.
- **Изучаемость:** единая логика работы во всех интерфейсах (командная строка, Qt GUI) и фиксированная последовательность этапов обработки обеспечивают быстрое освоение программы. Основным источником документации являются научные публикации, включая настоящую работу.
- **Управляемость:** поддержка пакетной обработки через командную строку и тонкая настройка параметров через графический интерфейс позволяют гибко управлять процессом обработки, адаптируя его под различные экспериментальные задачи.
- **Защищенность от ошибки пользователя:** система осуществляет валидацию входных данных; несовместимые опции блокируются на уровне интерфейса; при возникновении нештатных ситуаций выводятся диагностические сообщения, локализирующие причину сбоя.
- **Эстетика пользовательского интерфейса:** переход на кросс-платформенную библиотеку Qt в 2014 г. позволил создать современный графический интерфейс, сочетающий функциональную полноту с визуальной ясностью представления данных.

Подхарактеристика «**доступность**» не оценивалась, поскольку требования к поддержке специальных возможностей для лиц с ограничениями не предъявлялись.

За два десятилетия эксплуатации программа получила положительные отзывы от пользователей-физиков, отмечающих ее удобство для ежедневной работы на спектрометре ЮМО.

Надежность

Программа обеспечивает стабильное и предсказуемое выполнение функций в течение всего срока службы. Ее надежность подтверждается следующими реализованными подхарактеристиками.

- **Завершенность:** длительная эксплуатация (с 2003 года) на реальных экспериментальных данных подтверждает соответствие программы требованиям надежности.
- **Отказоустойчивость:** обработка заведомо некорректных или поврежденных данных (например, при сбоях на отдельных кольцах детектора) не приводит к аварийному завершению программы; пользователь получает диагностическую информацию для анализа и возможность скорректировать обработку (например, исключив проблемные кольца).
- **Восстанавливаемость:** в случае сбоя исходные данные не повреждаются; пользователь может повторно запустить обработку с измененными параметрами и получить корректный результат без потери информации.

Подхарактеристика «**готовность**» не оценивалась, поскольку SAS является обычным приложением, запускаемым по требованию; оценка готовности имела бы смысл при развертывании программы в качестве веб-сервиса.

Защищенность

Подхарактеристики защищенности (конфиденциальность, целостность, неподдельность, отслеживаемость, подлинность) не реализованы в программе SAS, поскольку требования к ограничению доступа, авторизации и аудиту не предъявлялись. Программа функционирует в изолированной вычислительной среде, не обрабатывает конфиденциальные данные и не требует механизмов защиты, выходящих за рамки стандартной политики безопасности ОИЯИ для научных установок.

Сопровождаемость

Легкость модификации для исправления дефектов, улучшений или адаптации к изменениям среды обеспечивается следующими подхарактеристиками.

- **Модульность:** архитектура реализует четкое разделение ответственности между классами (загрузка данных, обработка, визуализация).
- **Возможность многократного использования:** библиотека HTL разработана как независимый компонент с формализованным интерфейсом, что допускает ее применение в других программных проектах.
- **Анализируемость:** строгий конвейер обработки, в котором каждый класс отвечает за определенный этап, существенно упрощает диагностику и локализацию дефектов.
- **Модифицируемость:** физические константы и калибровочные параметры вынесены во внешние конфигурационные файлы, что позволяет вносить изменения без перекомпиляции программы. Изначально разработанная для двухдетекторной конфигурации спектрометра ЮМО, архитектура продемонстрировала достаточную гибкость для интеграции данных ПЧД без переработки вычислительного ядра.

- **Тестируемость:** наличие набора эталонных спектров для различных режимов измерений и строгая предсказуемость математических преобразований создают основу для регрессионного тестирования при внесении изменений в программный код.

Переносимость

Способность программы быть перенесенной из одной операционной или аппаратной среды в другую обеспечивается следующими подхарактеристиками.

- **Адаптируемость:** программа работает в различных операционных системах (Linux, Windows) и компилируется разными компиляторами (gcc, MSVC).
- **Устанавливаемость:** использование стандартного C++ и кросс-платформенных библиотек (Qt) обеспечивает единый процесс сборки и развертывания в разных средах.
- **Взаимозаменяемость:** новые версии SAS полностью замещают предыдущие, включая исходную версию 1992 года; обеспечена корректная обработка всех накопленных данных.

Таким образом, архитектурные решения SAS и их программная реализация, сформированные в процессе двадцатилетней эволюции (начиная с версии 2003 года), естественным образом достигают высокого уровня соответствия ключевым характеристикам качества, которые позже были систематизированы в современных стандартах. Соответствие стандарту ГОСТ Р ИСО/МЭК 25010-2015 (ISO/IEC 25010:2011) служит объективным критерием качества архитектуры с точки зрения ее долгосрочной устойчивости, поскольку этот стандарт явно выделяет и структурирует те самые характеристики ПО — прежде всего, сопровождаемость и переносимость, — от которых напрямую зависит его способность развиваться вместе с экспериментальной установкой. Двадцатилетняя эволюция SAS подтверждает, что изначальные архитектурные решения, соответствующие этим требованиям, обеспечивают не только текущую функциональность, но и долгосрочную жизнеспособность программного комплекса, его способность адаптироваться к будущим изменениям в методиках экспериментов.

Заключение

Двадцатилетняя история развития программы SAS демонстрирует успешный пример создания устойчивого ПО для научных экспериментов, соответствующего современным стандартам качества. Проведенный анализ позволяет предположить, что архитектурные принципы, реализованные в SAS, могут быть универсальными для создания устойчивого научного ПО. В отличие от многих узкоспециализированных решений SAS смогла пережить несколько циклов модернизации спектрометра ЮМО, адаптироваться к новым типам детекторов и форматам данных, оставаясь стандартным инструментом обработки на протяжении двух десятилетий.

Принципы разработки научного ПО

На основе ретроспективного анализа SAS можно сформулировать следующие рекомендации для создания устойчивого научного ПО.

1. **Анализировать унаследованные решения и предметную область.** Глубокое понимание предметной области через изучение унаследованного кода обеспечивает корректность новых реализаций — проявление *функциональной пригодности*. Наличие такого кода, как основы для регрессионного тестирования, напрямую повышает *сопровождаемость* программы.

2. **Проектировать классы, адекватно инкапсулирующие аппаратуру.** Создание пользовательских типов данных для оборудования обеспечивает модульность через изоляцию аппаратно зависимых параметров от логики обработки данных и модифицируемость при замене оборудования — аспекты *сопровождаемости*.
3. **Выстраивать обработку данных как конвейер.** Строгая последовательность четко разделенных этапов преобразований гарантирует функциональную полноту — аспект *функциональной пригодности*. Модульность этапов, их анализируемость и тестируемость по отдельности обеспечивают *сопровождаемость*. Изоляция потенциальных сбоев в пределах одного этапа повышает *надежность*.
4. **Выбирать внешние библиотеки с учетом долгосрочной поддерживаемости.** Стабильные библиотеки повышают *надежность*, но важно выбирать легкие решения, чтобы избежать зависимости от неподъемных фреймворков: это критически для *сопровождаемости* при ограниченных ресурсах.
5. **Выделять ключевые компоненты в собственные библиотеки.** Разработка библиотек с четкими интерфейсами повышает повторную используемость и анализируемость кода — аспекты *сопровождаемости*. Использование таких библиотек в разных проектах обеспечивает их всестороннее тестирование, что дополнительно повышает *надежность*.
6. **Реализовывать алгоритмы, обоснованные предметной областью.** Прямое использование аппаратных и физических характеристик эксперимента при построении алгоритмов исключает эмпирические приближения и обеспечивает корректность результатов — аспект *функциональной пригодности*.
7. **Обеспечивать долгосрочную доступность экспериментальных данных.** Эволюция к открытым форматам данных обеспечивает доступность данных для других инструментов анализа и предотвращает их блокировку в рамках одного программного комплекса. Поддержка старых форматов не позволяет потерять уже накопленные данные. Оба аспекта на практике обеспечивают *совместимость* системы.
8. **Предоставлять пользователю возможность работать без разработчика.** Полнофункциональные интерфейсы (CLI, GUI) позволяют пользователю самостоятельно выполнять все задачи обработки данных. Это повышает *удобство использования* и снижает нагрузку на разработчика. Единое вычислительное ядро гарантирует *надежность* результатов независимо от интерфейса.
9. **Сопровождать разработку научными публикациями.** Описание архитектурных решений и алгоритмов в научных статьях служит одновременно документацией, средством верификации и гарантией долгосрочной доступности, что напрямую влияет на *удобство использования*, *сопровождаемость* и *надежность* ПО.

Выработанные принципы и рекомендации могут быть применены в других проектах, связанных с обработкой экспериментальных данных. Следование им естественным образом обеспечивает соответствие ключевым характеристикам качества стандарта ГОСТ Р ИСО/МЭК 25010-2015 (ISO/IEC 25010:2011). Как показал анализ SAS, именно это соответствие создает основу для долгосрочной жизнеспособности научного ПО.

Современные задачи и направления развития

Среди актуальных задач, решаемых в рамках эволюции SAS, можно выделить:

- **уточнение процедуры нормировки:** исследование возможности использования длительно измеренных эталонных данных ванадия для повышения точности абсолютной нормировки;
- **улучшение алгоритмов усреднения:** реализация механизма автоматического отбрасывания статистических выбросов при усреднении спектров в пространстве q ;
- **оптимизация обработки данных ПЧД:** анализ влияния различных вариантов объединения ячеек ПЧД в кольцевые зоны на итоговое разрешение и статистическую точность данных.

Соответствие стандартам качества обеспечивает прочный фундамент для текущего развития системы.

Список литературы (References)

- ГОСТ Р ИСО/МЭК 25010-2015. Информационные технологии. Системная и программная инженерия. Требования и оценка качества систем и программного обеспечения (SQuaRE). Модели качества систем и программных продуктов. — М.: Стандартинформ, 2015. — Дата введения: 01.06.2016.
- GOST R ISO/IEC 25010-2015. Informacionnye tekhnologii. Sistemnaya i programmaya inzheneriya. Trebovaniya i ocenka kachestva sistem i programmogo obespecheniya (SQuaRE). Modeli kachestva sistem i programmnyh produktov [Information technology. System and software engineering. System and software Quality Requirements and Evaluation (SQuaRE). System and software quality models]. — Moscow: Standartinform, 2015. — Introduced: 01.06.2016 (in Russian).
- Кирилов А. С., Литвиненко Е. И., Астахова Н. В., Мурашкевич С. М., Петухова Т. Б.* Эволюция программного пакета SONIX для спектрометра ЮМО реактора ИБР-2 // Приборы и техника эксперимента. — 2004. — № 3. — С. 65–77.
- Kirilov A. S., Litvinenko E. I., Astakhova N. V., Murashkevich S. M., Petukhova T. B.* Evolution of the SONIX software package for the YuMO spectrometer at the IBR-2 reactor // Instruments and Experimental Techniques. — 2004. — Vol. 47, No. 3. — P. 334–345. — DOI: 10.1023/B:INET.0000032899.51622.1e (Original Russian paper: *Kirilov A. S., Litvinenko E. I., Astakhova N. V., Murashkevich S. M., Petukhova T. B.* Evolyutsiya programmogo paketa SONIX dlya spektrometra YUMO reaktora IBR-2 // Pribory i tekhnika eksperimenta. — 2004. — No. 3. — P. 65–77.)
- Кирилов А. С., Литвиненко Е. И., Астахова Н. В., Мурашкевич С. М., Петухова Т. Б., Юдин В. Е., Горделий В. И., Исламов А. Х., Куклин А. И.* Реорганизация программного пакета Sonix+ для работы с данными в форме списка событий // Письма в журнал физики элементарных частиц и атомного ядра. — 2024. — Т. 21, № 3. — С. 622–631.
- Kirilov A. S., Litvinenko E. I., Astakhova N. V., Murashkevich S. M., Petukhova T. B., Yudin V. E., Gordelii V. I., Islamov A. Kh., Kuklin A. I.* Reorganization of the Sonix+ software package to work with data in the event list form // Physics of Particles and Nuclei Letters. — 2024. — Vol. 21. — P. 502–507. — DOI: 10.1134/S1547477124700493 (Original Russian paper: *Kirilov A. S., Morkovnikov I. A., Murashkevich S. M., Petukhova T. B., Yudin V. E., Gordelii V. I., Islamov A. Kh., Kuklin A. I.* Reorganizatsiya programmogo paketa Sonix+ dlya raboty s dannymi v forme spiska sobyтий // Pis'ma v zhurnal fiziki elementarnykh chastits i atomnogo yadra. — 2024. — Vol. 21, No. 3. — P. 622–631.)
- Куклин А. И., Исламов А. Х., Ковалев Ю. С., Утробин П. И., Горделий В. И.* Оптимизация двухдетекторной системы малоуглового нейтронного спектрометра ЮМО для исследования нанообъектов // Поверхность. Рентгеновские, синхротронные и нейтронные исследования. — 2006. — № 6. — С. 74–83.
- Kuklin A. I., Islamov A. Kh., Kovalev Yu. S., Utrobin P. I., Gordelii V. I.* Optimizatsiya dvukhdetektornoy sistemy malouglavogo neytronnogo spektrometra YuMO dlya issledovaniya nanoob"ektov [Optimization of the two-detector system for the small-angle neutron spectrometer YuMO for nanoobject investigation] // Journal of Surface Investigation. X-ray, Synchrotron and Neutron Techniques. — 2006. — No. 6. — P. 74–83 (in Russian).
- Соловьев А. Г., Соловьева Т. М., Стадник А. В., Иванков О. И., Куклин А. И.* SAS. Программа для первичной обработки спектров малоуглового рассеяния. Версия 2.4. Описание и руководство пользователя // Сообщения ОИЯИ. — 2003. — P10-2003-86.

- Soloviev A. G., Solovjeva T. M., Stadnik A. V., Ivankov O. I., Kuklin A. I.* SAS. Programma dlya pervichnoy obrabotki spektrov malouglovogo rassyaniya. Versiya 2.4. Opisanie i rukovodstvo pol'zovatelya [SAS. The package for small-angle neutron scattering data treatment. Version 2.4. Long write-up and user's guide] // Communication of JINR. — 2003. — P10-2003-86 (in Russian).
- Blatt M.* DUNE as an example of sustainable open source scientific software development // arXiv preprint. — 2013. — <https://doi.org/10.48550/arXiv.1309.1783>
- Eisty N. U., Bernholdt D. E., Koufos A., Luet D. J., Mundt M.* Ten essential guidelines for building high-quality research software // arXiv preprint. — 2025. — <https://doi.org/10.48550/arXiv.2507.16166>
- FLTK: Fast light toolkit. — [Electronic resource]. — <https://www.fltk.org> (accessed: 13.02.2026).
- Franke D., Petoukhov M. V., Konarev P. V., Panjkovich A., Tuukkanen A., Mertens H. D. T., Kikhney A. G., Hajizadeh N. R., Franklin J. M., Jeffries C. M., Svergun D. I.* ATSAS 2.8: a comprehensive data analysis suite for small-angle scattering from macromolecular solutions // Journal of Applied Crystallography. — 2017. — Vol. 50, No. 4. — P. 1212–1225. — DOI: 10.1107/S1600576717007786
- Guinier A., Fournet G., Walker C. B., Yudowitch K. L.* Small-angle scattering of X-rays. — New York: Wiley, 1955. — 268 p.
- ISO/IEC 25010:2011. Systems and software engineering — Systems and software Quality Requirements and Evaluation (SQuARE) — System and software quality models. — Geneva: ISO, 2011.
- Kuklin A. I., Islamov A. Kh., Gordeliy V. I.* Two-detector system for small-angle neutron scattering instrument // Neutron News. — 2005. — Vol. 16, No. 3. — P. 16–18. — DOI: 10.1080/10448630500454361
- Manalastas-Cantos K., Konarev P. V., Hajizadeh N. R., Kikhney A. G., Petoukhov M. V., Molodenskiy D. S., Panjkovich A., Mertens H., Gruzinov A. Yu., Borges C., Jeffries C. M., Svergun D. I., Franke D.* ATSAS 3.0: expanded functionality and new tools for small-angle scattering data analysis // Journal of Applied Crystallography. — 2021. — Vol. 54. — P. 343–355. — DOI: 10.1107/S1600576720013412
- OpenGL — The industry's foundation for high performance graphics. — [Electronic resource]. — <https://www.opengl.org> (accessed: 13.02.2026).
- Ostanevich Yu. M.* Time-of-flight small-angle scattering spectrometers on pulsed neutron sources // Makromolekulare Chemie. Macromolecular Symposia. — 1988. — Vol. 15, No. 1. — P. 91–103. — DOI: 10.1002/masy.19880150107
- Petoukhov M. V., Franke D., Shkumatov A. V., Tria G., Kikhney A. G., Gajda M., Gorba C., Mertens H. D. T., Konarev P. V., Svergun D. I.* New developments in the ATSAS program package for small-angle scattering data analysis // Journal of Applied Crystallography. — 2012. — Vol. 45, No. 2. — P. 342–350. — DOI: 10.1107/S0021889812000332
- Petoukhov M. V., Konarev P. V., Kikhney A. G., Svergun D. I.* ATSAS 2.1 — towards automated and web-supported small-angle scattering data analysis // Journal of Applied Crystallography. — 2007. — Vol. 40, No. s1. — P. s223–s228. — DOI: 10.1107/S0021889807002853
- Qt — cross-platform software development for embedded & desktop. — [Electronic resource]. — <https://www.qt.io> (accessed: 13.02.2026).
- Soloviev A. G.* Gluplot: The gluplot plotting package. — [Electronic resource]. — <http://wwwinfo.jinr.ru/programs/jinr/lib/gluplot/indexe.html> (accessed: 13.02.2026a).
- Soloviev A. G.* HTL: Template library for one-dimensional histograms. — [Electronic resource]. — <http://wwwinfo.jinr.ru/programs/jinr/lib/htl/indexe.html> (accessed: 13.02.2026b).
- Soloviev A. G., Kutuzov S. A., Ivankov O. I., Kuklin A. I.* Primary data treatment software for position-sensitive detector of small-angle neutron scattering spectrometer in the isotropic pattern scattering case // EPJ Web of Conferences. — 2018. — Vol. 173. — P. 05015. — DOI: 10.1051/epjconf/201817305015

- Soloviev A. G., Litvinenko E. I., Ososkov G. A., Islamov A. Kh., Kuklin A. I.* Application of wavelet analysis to data treatment for small-angle neutron scattering // Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment. — 2003. — Vol. 502, No. 2. — P. 500–502. — DOI: 10.1016/S0168-9002(03)00481-9
- Soloviev A. G., Solovjeva T. M., Ivankov O. I., Soloviov D. V., Rogachev A. V., Kuklin A. I.* SAS program for two-detector system: seamless curve from both detectors // Journal of Physics: Conference Series. — 2017. — Vol. 848, No. 1. — P. 012020. — DOI: 10.1088/1742-6596/848/1/012020
- Soloviev A. G., Solovjeva T. M., Kuklin A. I.* SAS — package for small-angle neutron scattering data treatment. — [Electronic resource]. — <http://wwwinfo.jinr.ru/programs/jinrlib/sas/indexe.html> (accessed: 13.02.2026a).
- Soloviev A. G., Stadnik A. V., Islamov A. H., Kuklin A. I.* FITTER — fit a chosen theoretical multi-parameter function through a set of data points. — [Electronic resource]. — <http://wwwinfo.jinr.ru/programs/jinrlib/fitter/indexe.html> (accessed: 13.02.2026b).
- Zech P., Pekaric I.* Sustaining research software: A fitness function approach // arXiv preprint. — 2025. — <https://doi.org/10.48550/arXiv.2509.10085>