

Восстановление треков заряженных частиц по данным электромагнитного калориметра

Нехаенко Павел

Кафедра математического анализа Яргу им. П.Г. Демидова

2025

Введение

В данной работе рассматривается задача восстановления трёхмерного распределения энергособытий в калориметре по двум двумерным проекциям.

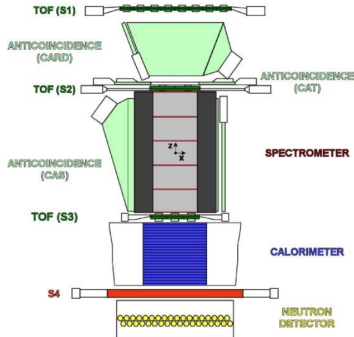
Такие задачи возникают в физике при анализе прохождения заряженных частиц через многослойные детекторы.

Мы будем решать обратную задачу: по двум проекциям — вдоль осей X и Y — попытаться восстановить полное 3D-распределение.

Ключевой целью является определение направления движения частицы и характера её взаимодействия с веществом детектора.

Калориметр PAMELA

- Калориметр в составе эксперимента PAMELA (Payload for Antimatter Matter Exploration and Light-nuclei Astrophysics) предназначен для регистрации и анализа частиц, проходящих через детектор.



Постановка задачи

- $x \in \mathbb{R}^{Z \times Y \times X}$ — трёхмерное распределение энергии по слоям.
- Проекции:

$$b_{z,y}^{(x)} = \sum_x x_{z,y,x} \quad (\text{проекция по оси } X)$$

$$b_{z,x}^{(y)} = \sum_y x_{z,y,x} \quad (\text{проекция по оси } Y)$$

- Объединяя проекции в вектор b , получаем:

$$Ax = b$$

- Добавляем регуляризацию и неотрицательность:

$$\min_{x \geq 0} \|Ax - b\|^2 + \lambda R(x)$$

Сложности задачи

- Недоопределённость: уравнений меньше, чем переменных.
- Проекции независимы, решение может не существовать.
- Сильное влияние шума и вторичных частиц.
- Требуется устойчивая численная процедура.

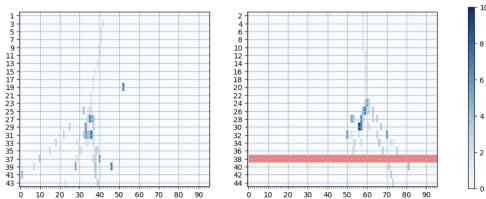


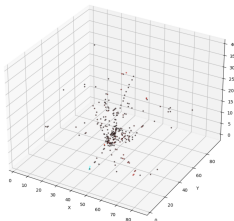
Рис. 1: Проекции энерговывделений

Подход к решению

- Формулируем задачу как:

$$\min_{x \geq 0} \|Ax - b\|^2 + \lambda \|Dx\|^2$$

- A — матрица проекции из 3D в 2D (b — наблюдаемые проекции).
- D — матрица сглаживания по оси Z .
- Ограничение $x \geq 0$ отражает физический смысл: энергия не может быть отрицательной.
- Выход: восстановленный объем x и направление трека.



Подход к решению

Для работы были созданы следующие подготовительные функции:

- `get_volume_for_event` — собирает объём из исходных координат.
- `compute_projections` — суммирует объём по X и Y (b).
- `build_projection_matrix` — строит матрицу A .
- `smoothness_matrix` — строит регуляризатор D .
- `compute_track_angles` — восстанавливает направление трека по центрам масс.
- `get_true_angles` — извлекает истинные углы θ_0 , ϕ_0 .
- `clean_min_energy` — зануляет ячейки с минимальной энергией для удаления шума.

L2-регуляризация по оси Z

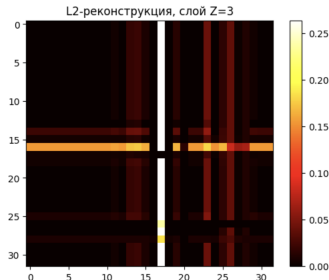
- Регуляризация по Z сглаживает значения между соседними слоями:

$$R(x) = \|Dx\|^2$$

- Матрица D содержит разности между слоями:

$$D_i = e_i - e_{i+1}$$

- Подходит для простых случаев, но размывает трек.

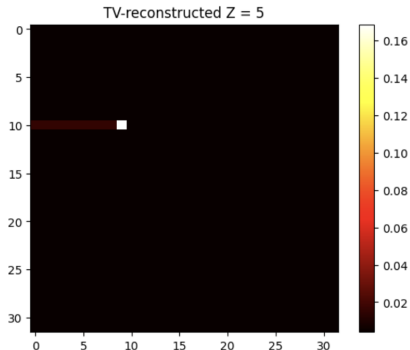


TV-регуляризация

- TV-регуляризация сохраняет резкие границы:

$$R(x) = \|Dx\|_1$$

- В отличие от L2, не размывает трек.
- Лучше восстанавливает локальные пики и форму траектории.



Масштабирование: от прототипа к реалистичному объёму

- Начинали с малого объёма: $8 \times 32 \times 32$ (8192 переменных).
- Далее: $16 \times 64 \times 64$ (65536 переменных).
- Финальная цель: $44 \times 96 \times 96$ (405504 переменных).
- Все размеры поддерживаются одним и тем же кодом.

```
Numerical solver
(CVXPY) May 25 11:13:47 PM: Invoking solver OSQP to obtain a solution.

OSQP v1.0.0 - Operator Splitting QP Solver
(c) The OSQP Developer Team

problem: variables n = 1684363, constraints m = 2803211
nnz(P) + nnz(A) = 7599670
settings: algebra = Built-in,
OSQPInt = 4 bytes, OSQPFloat = 8 bytes,
linear system solver = QDLDL v0.1.8,
eps_abs = 1.0e-05, eps_rel = 1.0e-05,
eps_prim_inf = 1.0e-04, eps_dual_inf = 1.0e-04,
rho = 1.00e-01 (adaptive: 50 iterations),
sigma = 1.00e-06, alpha = 1.60, max_iter = 10000
check_termination: on (interval 25, duality gap: on),
time_limit: 1.00e+10 sec,
scaling: on (10 iterations), scaled_termination: off
warm_starting: on, polishing: on,

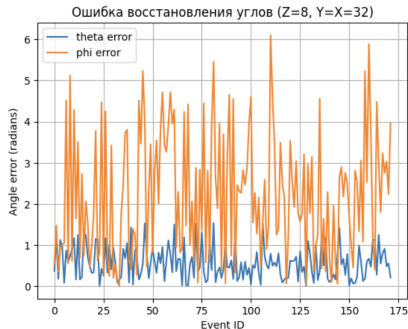
iter  objective    prim res    dual res    gap    rel kkt    rho    time
1    -9.5987e+04    8.00e+00    7.20e-01    -9.59e+04    8.00e+00    1.00e-01    1.28e+03s
50   6.5378e-05    3.98e-03    3.05e-07    -6.75e-04    3.98e-03    1.77e+00    1.79e+03s
200  1.4881e-02    3.34e-03    6.71e-04    -9.74e-03    3.34e-03    1.77e+00    4.63e+03s
350  2.5045e-02    6.63e-04    6.73e-04    -1.09e-03    6.73e-04    2.70e-01    5.62e+03s
Solver interrupted
```

Анализ ошибок восстановления направления

- После восстановления объёма оцениваем направление трека:

$\theta, \phi \leftarrow$ регрессия центра масс по слоям

- Сравниваем с истинными углами θ_0, ϕ_0 , известными из модельных данных.
- Метрики: $|\theta - \theta_0|, |\phi - \phi_0|$ для каждого события.



Ограничения и вычислительные ресурсы

- Размер задачи растёт быстро:
 - $8 \times 32 \times 32 \rightarrow 8192$ переменных;
 - $44 \times 96 \times 96 \rightarrow 405504$ переменных.
- Матрицы A , D разреженные, но всё равно требуют памяти.
- При полном размере решение занимает 40+ минут на ноутбуке.
- TV-регуляризация особенно ресурсоёмка.
- Возможность переноса на HPC/кластер.

```
Numerical solver
(CVXPY) May 25 11:13:47 PM: Invoking solver OSQP to obtain a solution.
=====
OSQP v1.0.0 - Operator Splitting QP Solver
(c) The OSQP Developer Team
=====
problem: variables n = 1604363, constraints m = 2803211
nnz(P) + nnz(A) = 7599670
settings: algebra = Built-in,
OSQPInt = 4 bytes, OSQPFLOAT = 8 bytes,
linear system solver = QDLDL v0.1.6,
eps_abs = 1.0e-05, eps_rel = 1.0e-05,
eps_prim_inf = 1.0e-04, eps_dual_inf = 1.0e-04,
rho = 1.00e-01 (adaptive: 50 iterations),
sigma = 1.00e-06, alpha = 1.60, max_iter = 10000,
check_termination: on (interval 25, duality gap: on),
time_limit: 1.00e+10 sec,
scaling: on (10 iterations), scaled_termination: off
warm starting: on, polishing: on.
iter objective prim res dual res gap rel kkt rho time
1 -9.5907e+04 8.00e+00 7.20e-01 -9.59e+04 8.00e+00 1.00e-01 1.28e+03s
50 6.5378e-05 3.98e-03 3.05e-07 -6.75e-04 3.98e-03 1.77e+00 1.79e+03s
200 1.4881e-02 3.34e-03 6.71e-04 -9.74e-03 3.34e-03 1.77e+00 4.63e+03s
350 2.5845e-02 6.63e-04 6.73e-04 -1.09e-03 6.73e-04 2.70e-01 5.62e+03s
Solver interrupted
```

Выводы и перспективы

- Разработан прототип восстановления 3D-распределения по двум проекциям.
- Реализовано: построение матриц, регуляризация, решение задачи, извлечение трека.
- Проведено масштабирование: от $8 \times 32 \times 32$ до $44 \times 96 \times 96$.
- Проведено сравнение L2 и TV-регуляризации, анализ ошибок.
- Возможные расширения:
 - Использование других регуляризаторов;
 - Ускорение за счёт инициализации или эвристик;
 - Построение батч-реконструкции и интеграция в пайплайн.