Менторский проект

Никита Косогоров
Московский физико-технический институт
Лаборатория релятивистской астрофизики
https://rellab.mipt.ru/

I. Радиопеременность, кинематика и геометрия релятивистских джетов

A. Простая подгонка данных

В случае линейной зависимости функцию потерь можно записать в виде: $L(a,b) = \sum_i (y_i - f(x_i))^2 = \sum_i (y_i - (ax_i + b))^2$. Дифференцируя ее по параметрам a,b и приравнивая производные к нулю, можно получить известные формулы для метода наименьших квадратов.

$$\begin{cases} \frac{\partial L}{\partial a} = 0 \to a = \frac{n\sum_i x_i y_i - \sum_i x_i \sum_i y_i}{n\sum_i x_i^2 - \left(\sum_i x_i\right)^2} \\ \frac{\partial L}{\partial b} = 0 \to b = \frac{\sum_i y_i - a\sum_i x_i}{n} \end{cases}$$

В данном задании функция, которая описывает данные, зависит от одного параметра $f(x) = ax^2$. Нужно получить формулу на a аналитически с помощью дифференцирования функции потерь $L(a) = \sum_i \left(y_i - ax_i^2\right)^2$. Дальше по полученной формуле посчитать a. После найти параметр a с помощью curve_fit из scipy, и потом сравнить полученные результаты. Кроме этого, нарисовать график данных и на нем же график зависимости $y = a_0 x^2$,где a_0 – значение, которое получилось с помощью curve_fit. Данные находятся в текстовых файлах: x_data.txt – значения x, y_data_quad.txt – значения y.

Также есть зависимость $y = ae^{-b\sin(x)} + cx$. Надо найти коэф a, b и c также с помощью scipy. Данные находятся в текстовых файлах: x_data.txt – значения x, y_data_nonlinear.txt – значения y. Внизу показан график с данными (Рис. 1), нужно построить график подогнанной зависимости.

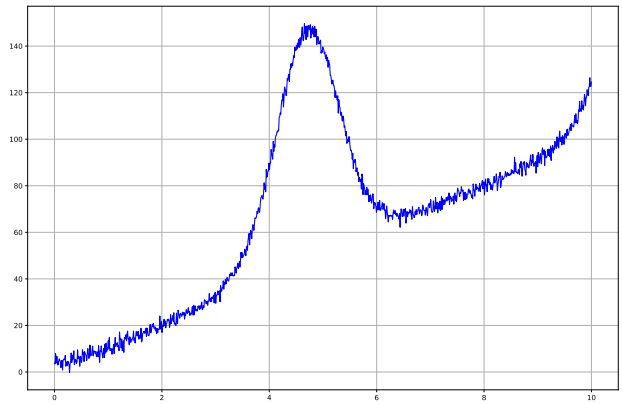

Рис. 1. График данных для первого задания

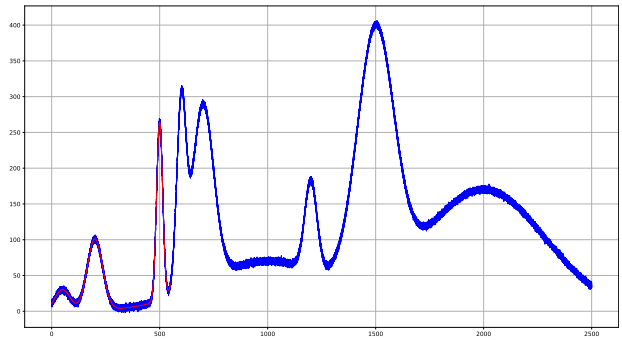

Рис. 2. График данных (синим) для второго задания и подогнанная зависимость (красным) для части данных

B. Подгонка гауссиан и спектра синхротронного источника

1. *Подгонка n гауссиан*

Есть функция, которая равна сумме n гауссиан, $y = \sum_{i=0}^{n-1} a_i e^{-\left(\frac{x-b_i}{c_i}\right)^2}$. Найти 3n параметров – a_i, b_i и c_i. Построить графики данных из файлов и полученной функции. x_data, y_data – прилагаемые данные для функции. Внизу на Рис. 2 показаны график данных, а также часть подгоняемой зависимости.

2. *Спектр однородного синхротронного источника*

Спектр однородного синхротронного источника может быть описан с помощью зависимости потока от частоты:

$$I(\nu) = I_{\mathrm{m}} \left(\frac{\nu}{\nu_{\mathrm{m}}}\right)^{\alpha_{\mathrm{thick}}} \frac{1 - exp(-\tau_{\mathrm{m}}(\nu/\nu_{\mathrm{m}})^{\alpha_{\mathrm{thin}} - \alpha_{\mathrm{thick}}})}{1 - exp(-\tau_{\mathrm{m}})},$$

где

$$\tau_{\mathrm{m}} = \frac{3}{2}\left(\sqrt{1 - \frac{8\alpha_{\mathrm{thin}}}{\alpha_{\mathrm{thick}}}} - 1\right)$$

α_{thick} положить равным 2.5. Есть три текстовых файла – freq.txt (значения частоты), fluxdensity.txt (значения плотности потока) и errflux.txt (ошибки в потоке).

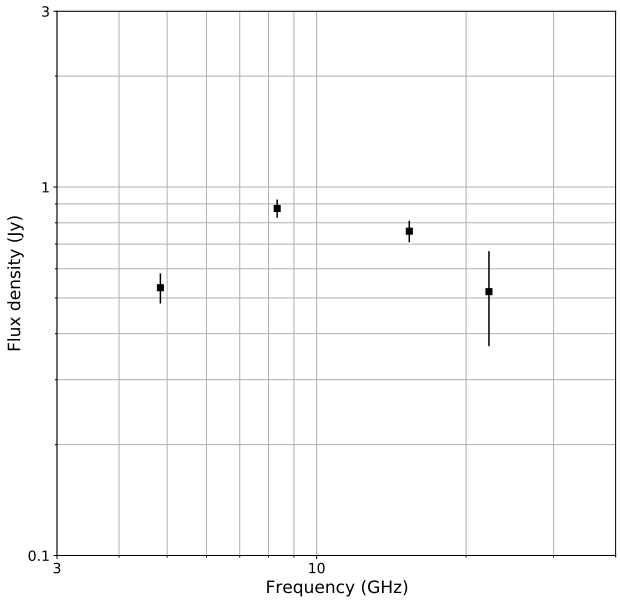

Рис. 3. Данные для моделирования спектра синхротронного источника

Найти три параметра – $\alpha_{\rm thin}$, $I_{\rm m}$ и $\nu_{\rm m}$ в двух случаях: без учета ошибок в потоке и с учетом. Построить графики полученных зависимостей в логарифмическом масштабе по обоим осям, нанести сами точки на график, а также их ошибки. Напоминаю, что ошибки учитываются с помощью весов в функции потерь $L = \sum_i \left(\frac{y_i - f(x_i)}{\sigma_i}\right)^2$ и, как правило, подаются в функцию в качестве дополнительного параметра (в случае curve_fit – это sigma). На Рис. 3 показаны точки с данными.

C. Радиопеременность излучения

Прочитать первые два пункта статьи Doppler factors, Lorentz factors and viewing angles for quasars, BL Lacertae objects and radio galaxies, – Intro и Data and the method https://arxiv.org/abs/0811.4278.

Данные – три csv файла, в каждом из которых даны зависимости потока от времени и еще ошибки потока (пример графика для одного объекта на Рис. 4). Задание промоделировать данные экспоненциальными вспышками, описанными в статье. Придумать какой-нибудь критерий для оценки того, насколько хорошо модельная функция подгоняет данные (например, метрика MSE). Попробовать подогнать с разным колвом вспышек, а также немного изменить вид функции (например, в знаменателе формулы стоит 1.3τ, можно попробовать взять вместо 1.3 что-нибудь другое). С помощью взятого критерия найти какая модель лучше всех описывает данные. Используя полученные модели посчитать Doppler factor D_{var} (eq.(3)) для каждой вспышки.

Наблюдения проводились на 15 ГГц. Для поиска расстояния можно воспользоваться онлайн калькулятором: Cosmology calculator.
Красные смещения:
J0854+2006 $z = 0.306$
J0400+0550 $z = 0.761$
J1025+1253 $z = 0.663$

D. Две 2D гауссианы

В этом задании надо подогнать две двумерные гауссианы вида: $\mathrm{A}e^{-\frac{(x_0-x)^2}{\sigma^2}-\frac{(y_0-y)^2}{\sigma^2}}$. Из данных – два файла image_data.txt и опорный jupyter notebook. Внизу на Рис. 5 показан график данных в двумерии цветом и в трехмерии. Также показана получаемая модель в 3D.

Подсказака: амплитуды A для начального приближения можно взять равными 10000, координаты центров гауссиан - из картинки, сигму σ равной 50.
Для подгонки можно использовать curve_fit. Также полезными функциями могут быть np.ravel(), np.mgrid(), np.shape()

E. Кинематика и геометрия 0851+202

Задание описано в прилагаемом jupyter notebook и состоит из пяти частей. В первой получаем изображение квазара 0851+202 на нескольких эпохах в Python (для подробностей как это сделать можно посмотреть на аналогичное задание из section II A). Во второй части моделируем излучение $I(\alpha, \theta)$ с помощью двумерных гауссиан. Также для проверки можно промоделировать структуру в difmap (см. подробности как это делать в section II F). В третьей части оцениваем скорость движущейся компоненты. В четвертой находим релятивисткий Доплер-фактор (см. section I C). В пятой части оцениваем геометрию и кинематику с помощью формул:

$$\Gamma_{\rm var} = \frac{\beta_{\rm app}^2 + D_{\rm var}^2 + 1}{2D_{\rm var}}$$

$$\theta_{\rm var} = \arctan \frac{2\beta_{\rm app}}{\beta_{\rm app}^2 + D_{\rm var}^2 - 1}$$

II. Картографирование квазаров

A. Построение изображения квазара в Python

В данном задании необходимо считать fits файл с данными наблюдений квазара 1226+023 в Python и построить его изображение. Данные – сам fits файл, а также опорный jupyter notebook с подробностями.

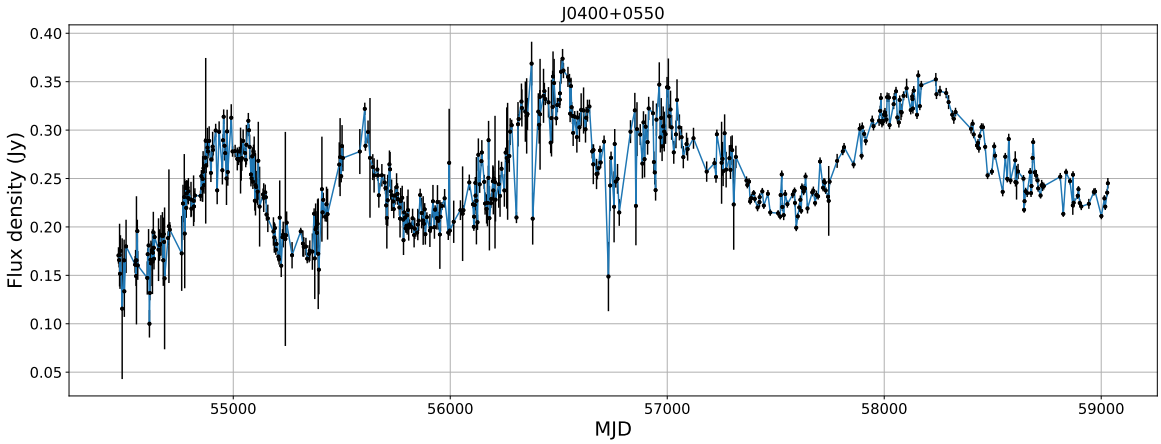

Рис. 4. Кривая блеска для J0400+0550

B. Дискретное Фурье-преобразование

1. *Быстрое Фурье-преобразование*

В jupyter notebook дана одномерная функция амплитуды от времени. Необходимо написать функцию, которая считает дискретное преобразование Фурье. Формулу, которая его задает, можно найти, например, здесь: Discrete Fourier transform. После посмотреть сколько по времени она работает %timeit DFT(x) и сравнить с реализованным быстрым преобразованием Фурье %timeit np.fft.fft(x).

2. *Фурье-преобразование изображений со сменой амплитуд и фаз*

Во второй части можно взять две любые картинки. Считать их в массив. Привести их к одинаковому размеру. Сделать из них черно-белые картинки, если взяли цветные (пример как это можно реализовать в ноутбуке).

Дальше сделать Фурье преобразование обоих картинок (можно использовать быстрое Фурье преобразование numpy.fft.fft2, можно попробовать найти обычное или реализовать самим). После поменять фазу и амплитуду – в каждом пикселе получается комплексное число, которое можно представить в виде $Ae^{i\phi}$, где A - амплитуда, ϕ - фаза. Нужно оставить амплитуды у этих двух изображений (под этим я имею в виду амплитуды всех комплексных чисел, которые есть в массиве), а фазы поменять местами. После надо сделать обратное преобразование Фурье numpy.fft.ifft2. Нарисовать изображения для полученных данных (матрица будет комплексной, поэтому попробуйте построить ее модуль).

Дополнительно почитать и ознакомиться со свертками: Convolution

C. Установка программного пакета difmap

Скачать и установить difmap. Более детально, что нужно писать в командной строке:
ftp ftp.astro.caltech.edu
login: anonymous
password: **hit enter**
cd pub/difmap
get difmap2.5e.tar.gz
tar xzf difmap2.5e.tar.gz
cd uvf_difmap/

Дальше заходим в READ.ME и читаем как устанавливать. Есть шанс, что у вас не будет чего-то нужного для установки (например, фортрановского компилятора). В этом случае просто гуглите, как установить. Если возникает какая-то ошибка при установке, то после исправления нужно начинать установку заново, не забыв перед этим сделать ./clean. Некоторые полезные ссылки с гайдами по установки: brandeisastro installing difmap on linux or mac, youtube tutorial – Installing DIFMAP, pgplot installation.

D. Картографирование в difmap

Полезные презентации: Gentle_intro_to_interferometry, Imaging

Скачайте файл по ссылке 3C_273_vis_data (если автоматически не скачивается, можно использовать в терминале wget *ссылка)
Запустите difmap в директории, где находится скачанный файл.
В нем напишите команду observe имя_файла. Вы увидите некоторую информацию о данных. Чтобы узнать подробности использования функции, напишите help observe (так можно делать с любой функцией).

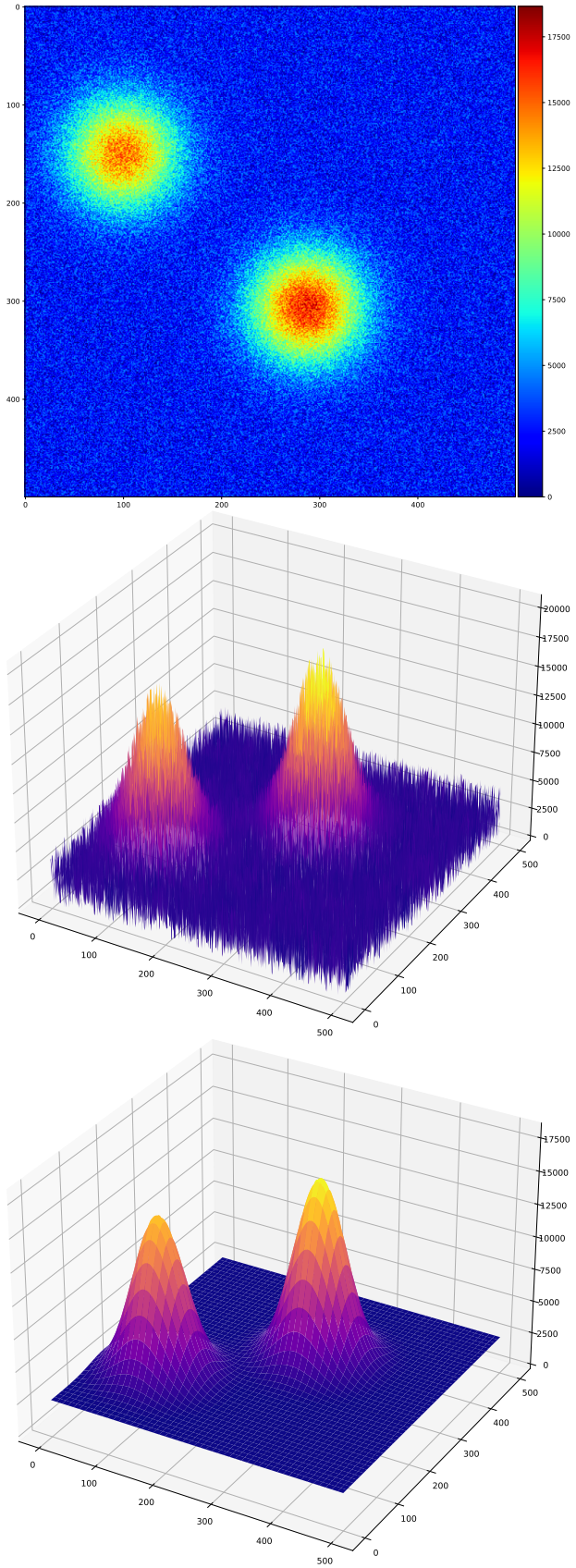

Рис. 5. Данные для моделирования двух 2D гауссиан (две верхние картинки). Внизу – получаемая модель.

Выберем нужную поляризацию (полную интенсивность): select I

Посмотрим на зависимость функции видности от uv-радиуса V($\sqrt{u^2+v^2}$): radplot (если не появилось окно, напишите dev /xw) Наведите курсор на окно и нажмите H. В терминале вы увидите, что можно посмотреть в окне, нажимая различные кнопки. Например, наведите на окно и нажмите 3 и левая кнопка мыши, рисунок поменяется, будет показана и фаза и амплитуда. Или нажимайте на кнопку N, наведя на окно, будут выделяться разные телескопы. Попробуйте разные команды. Чтобы выйти обратно в терминал, наведите на окно и нажмите на правую кнопку мыши (работает и дальше с другими всплывающими окнами после использования других функций).

Посмотрите на uv-покрытие: uvplot

Зададим размер изображения (количество пикселей и сколько mas в одном пикселе): mapsize 1024,0.1

Посмотрим на dirty beam: maplot beam. Чтобы посмотреть на какую-то область изображения более близко, нажмите Z и выделите нужный квадрат курсором с помощью двух кликов левой кнопкой. Чтобы сохранить данную область при новом всплывании окна, нажмите K. Чтобы отменить и вернуться обратно ко всей области два раза нажмите Z.

Посмотрим на "грязную"карту: maplot. Приблизьте в центр. После выделите квадратом яркую часть изображения с сигналом, нажимая на левую кнопки мыши и выделяя квадрат. Должно получиться примерно как на Рис. 6. Обратите внимание на шкалу внизу, она показывает цветовую шкалу интенсивности. Видно, что максимум интенсивности в районе 3 Jy/beam. Выйдите в терминал.

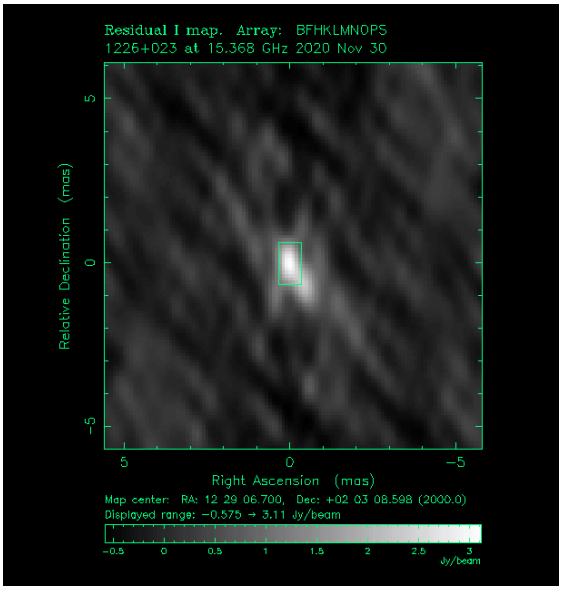

Рис. 6. Первое выделение яркой области квазара для последующего CLEAN-алгоритма

Переходим к CLEAN-алгоритму. Выбираем вес uvw 2,-1.
Проводим 100 итераций алгоритма clean 100,0.03
Заходим обратно на карту maplot. Нажимаем M, внутри выделенной области появятся дельта компоненты. Среди них могут быть красные (означает с отрицательным потоком), их желательно избегать на начальных стадиях. Самая яркая область изображения сместится. Выделяем теперь ее. Может понадобиться удалить какое-нибудь зеленое окно, если оно плохо получилось. Для этого можно нажать D, при этом исчезнет то, которое находится ближе всего к курсору. Выходим в терминал и пишем clrmod true,true; uvw 2,-1; clean 400,0.03. Снова заходим в карту и выделяем, если появилась новая яркая область, используя clrmod true,true; uvw 2,-1; clean 400,0.03. Если ничего не появляется, то добавляем еще несколько интераций, при этом изменяя вес данных clrmod true,true; uvw 2,-1; clean 400,0.03; uvw 0,-2; clean 500.
Повторяем пока не останется один шум (чтобы понять, что остался один шум, можно посмотреть на шкалу интенсивности внизу, максимальные значения должны стать значительно меньше 3 Jy/beam, около 0.05 Jy/beam, не появляются новые детали и излучение выглядит относительно однородно)
В терминале пишем команду maplot clean. Это модельная карта джета. На карте в подписи помимо максимума интенсивности и показанных контуров, также есть параметры диаграммы направленности. Их также можно посмотреть, подав в терминале команду restore. Кроме того, можно задать свои параметры диаграммы направленности и посмотреть на соответствующее изображение (например, можно задать круглый beam restore 0.9,0.9,0).
Сохраним то, что у нас получилось save 1226. Один из файлов будет fits изображением. Возьмите его и получите изображение в питоне, как в первом задании. Для того, чтобы сохранить изображение в других форматах см. section II E. Изображение, которое могло получиться на Рис. 7.

Выйти из дифмапа можно с помощью quit.

E. Самокалибровка

Берем тот же файл из прошлого задания: 3C273_data_2020
Выполняем все шаги из прошлого задания до сохранения файла. Делаем фазовую самокалибровку selfcal
Дальше выполняем шаги CLEAN-алгоритма, которые были до самокалибровки clrmod true,true;uvw 2,-1;clean 400,0.03;clrmod true,true;uvw 0,-2;clean 800.
Пытаемся увеличить количество итераций и если появились новые яркие области, выделить их.
Теперь делаем фазовую и амплитудную самокалибровку (1e6 - solution interval)
selfcal
selfcal true,true,1e6

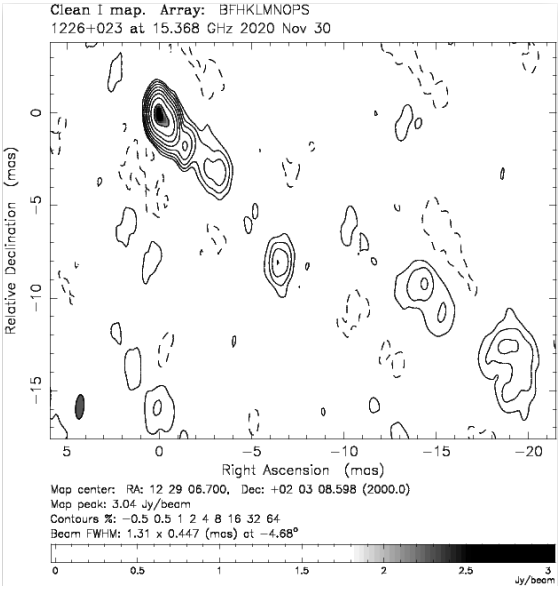

Рис. 7. Изображение 1226+023

selfcal
Дальше снова приступаем к CLEAN-алгоритму, и повторяем все шаги с фазовой самокалибровки, постепенно уменьшая solution interval в амплитудной самокалибровке (после 1e6, постепенно уменьшаем и берем значения: 120, 60, 20, 10, 5, 2, 0.5)
После каждой амплитудной самокалибровки (selfcal true,true,your_solint) можно посмотреть с помощью corplot на Gain. Постепенно можно будет увидеть, что модуль немного отличается от 1. Так как данные хорошие, то сильных отличий от 1 не должно быть (фаза также должна быть примерно нулевая). Если бы значения где-то сильно отличались от этих, то это могло свидетельствовать о том, что данные плохие и их нужно флагировать.
Смотрим на финальную карту maplot clean. Если нужно изменить уровни контуров, то можно воспользоваться командой loglev *параметры_контуров (напоминаю, что параметры которые подаются на вход для функции можно посмотреть с помощью help)
Сохраняем все и выходим. Также можно отдельно сохранить изображение не в fits формате, а в каком-нибудь другом. Например:
dev 1226_U_2020.ps/VPS
mapl cln
dev /XW
Теперь на выбор можно получить изображение этого квазара в 2010 году 3C273_data_2010 или другого BL_Lac

F. Моделирование стуктуры

Берем файл 3C_273_vis_data. Считываем с помощью observe. Выберем нужную поляризацию (полную

интенсивность): select I. Зададим размер изображения: mapsize 1024,0.1.

Для того, чтобы промоделировать структуру с помощью гауссиан, делаем maplot и на карте находим яркое излучение. Теперь вместо выделения квадратов, в которых мы искали CLEAN компоненты, мы нажимаем на N и после на левую кнопку мыши. При этом ставим курсор в центр яркой детали, дальше отводим курсор. При этом появляется синяя ось, нажимаем на левую кнопку, когда доходим до края яркой детали, и после жмем на N. После будет примерно как на Рис. 8. Так мы выделили одну гаусс-компоненту,

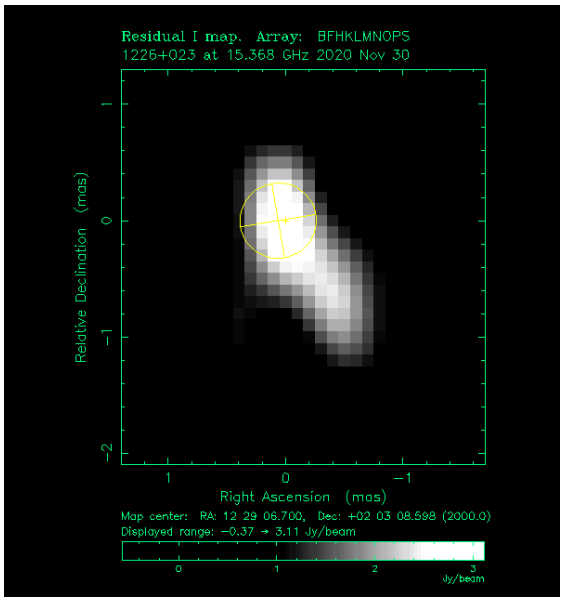

Рис. 8. Первая гауссиана для самой яркой детали квазара

теперь выходим в терминал, нажимая на окне с квазаром на правую кнопку мыши и подаем команду: modelfit 100 для подгона параметров компоненты. После снова заходим на изображение и продолжаем выделять гауссианы в ярких деталях квазара и выходить в терминал, подавая modelfit 100. Можно обратить внимание, что в терминале будут написаны значения χ^2_{red} модели. При этом при увеличении количества гауссиан, он как правило должен уменьшаться. После нескольких итераций может получится что-то наподобие изображений на Рис. 9 и Рис. 10. Для того, чтобы посмотреть параметры компонент можно подать команду edmodel и увидеть их параметры в терминале. Или же можно сохранить в текстовый файл wmod *название_файла. Также можно глянуть на модель посмотрев на V(uv-radius) с помощью команды radplot (модель показана красным цветом).

G. Поляризация

Про используемые в этом задании параметры Стокса можно прочитать, например, здесь:

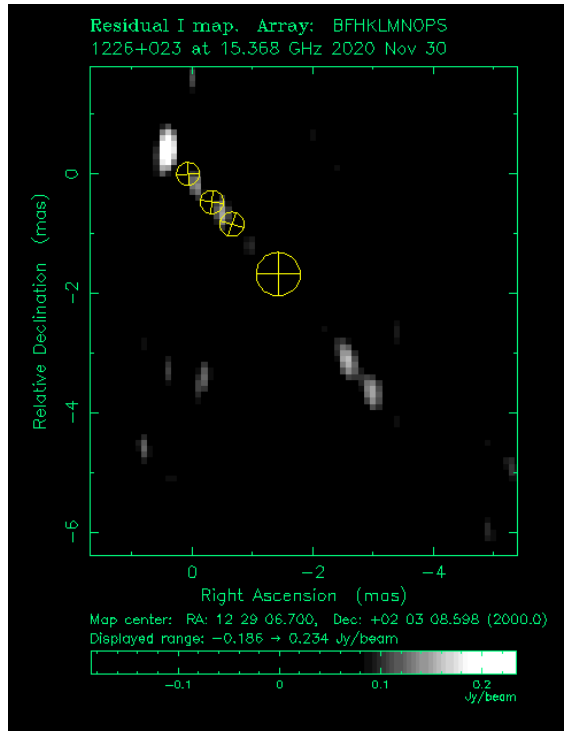

Рис. 9. Промежуточные гауссианы при моделировании

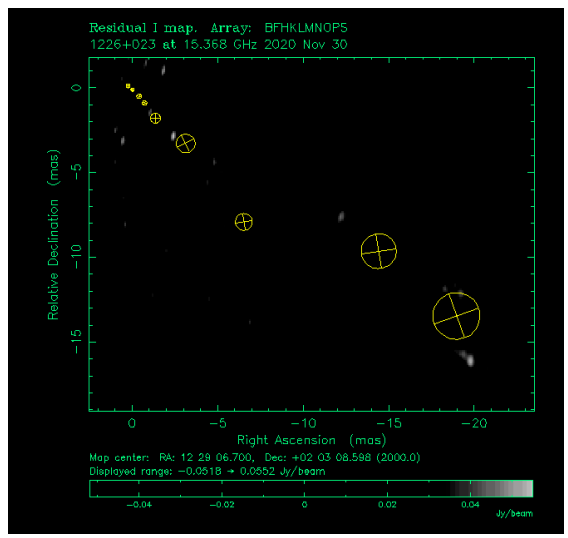

Рис. 10. Финальные гауссианы для моделировнаия структуры источника

Stokes_parameters. Из данных три fits файла с данными для I,U,Q параметров Стокса для квазара 1328+307 на частоте 1.5 ГГц. Необходимо их считать, как в section II A. После построить карту полной интенсивности. Размер одного пикселя – 0.7 mas. Комплексаня поляризация задается:

$$\Pi = Q + iU = Pe^{2i\chi}\,,$$

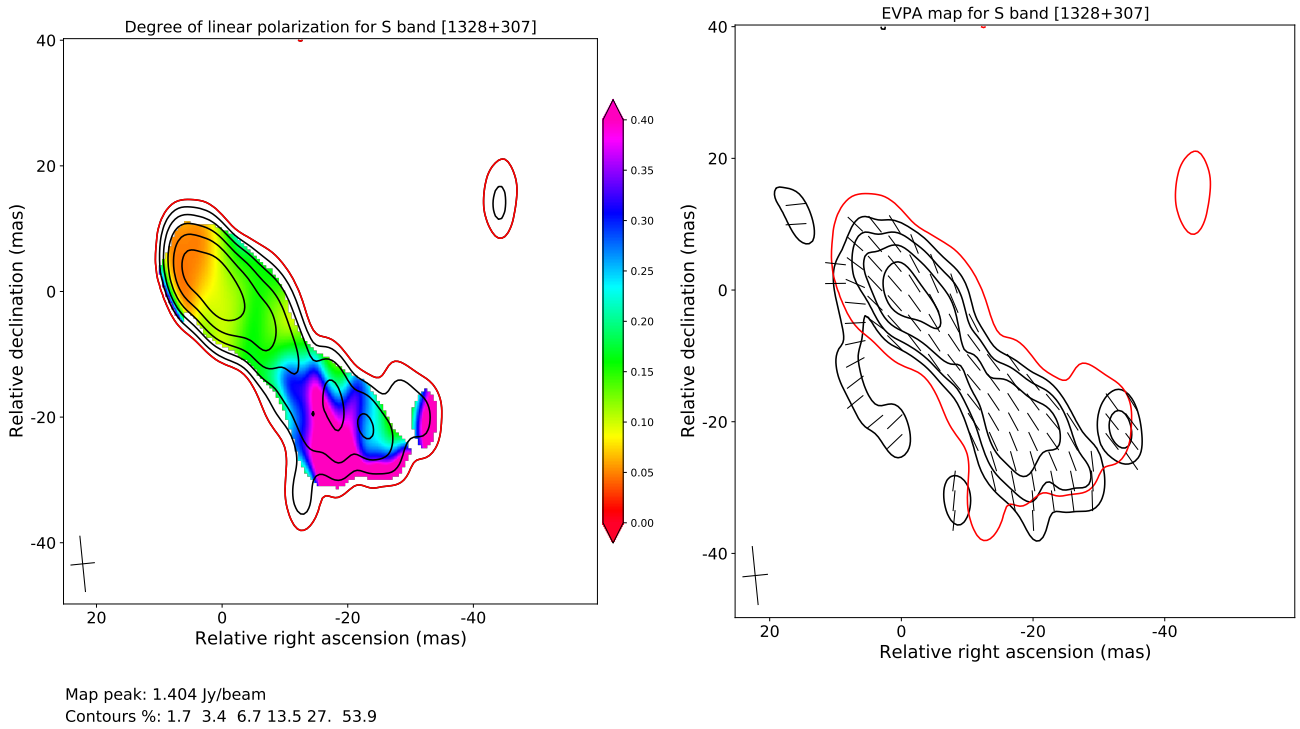

Рис. 11. Карты степени и направления поляризации для 1328+307 на 2.3 ГГц.

где P – модуль линейной поляризации, а χ – EVPA (electric vector position angle).

В первой части необходимо построить карту степени линейной поляризации m, которая задается формулой:

$$m = \frac{P}{I} = \frac{\sqrt{U^2 + Q^2}}{I}$$

Необходимо показывать значения лишь там где P превосходит $3\sigma_P$. При этом σ_P считается аналогично σ_I (см. ноутбук из section II A). На Рис. 11 слева показан пример такой карты для этого же источника только на другой частоте (с наложенными контурами полной интесивности). Во второй части необходимо построить карту абсолютного EVPA (χ_0). Угол χ_0 задается:

$$\chi_0 = \chi - \Delta\chi = \frac{1}{2}\arctan\frac{U}{Q} - \Delta\chi,$$

где $\Delta\chi$ – поправка, связанная с калибровкой абсолютной поляризации. В данном задании ее можно принять равной -67 градусам. Значения направления можно показывать, например, короткими линиями, как на Рис. 11 справа. Необходимо показывать направления тоже только там, где P превосходит $3\sigma_P$. На карте на Рис. 11 наложены контуры линейной поляризации P.

Данные, используемые в заданиях

Все используемые в заданиях файлы с данными, а также jupyter notebooks можно найти по ссылке в google drive: link_with_all_materials.