

Методическое пособие для первокурсников ФФПФ МФТИ
по курсу «Проектная деятельность»
для проекта «Анализ колебаний солнечных корональных петель»

**Методы фитирования объектов с изображений с
помощью языка программирования МАТЛАВ,
корональные солнечные петли**

Март 2019

Содержание

I	Теоретическая часть	2
1	Что такое корональные петли?	2
II	Практическая часть	4
2	Загрузка изображений в МАТЛАВ для работы	4
3	Фитирование данных	5
III	Задачи для самостоятельной работы	7

Вашим самым близким другом при работе с МАТЛАВ должен стать раздел HELP, для поиска по которому выделите ваш запрос и нажмите кнопку F1. В хэлпе максимально ёмко описаны все команды и приведены примеры их использования. Также настоятельно рекомендую попутно с чтением методички держать открытой программу и вбивать туда всё, что вы видите в методичке, чтобы самостоятельно прочувствовать результат.

Часть I

Теоретическая часть

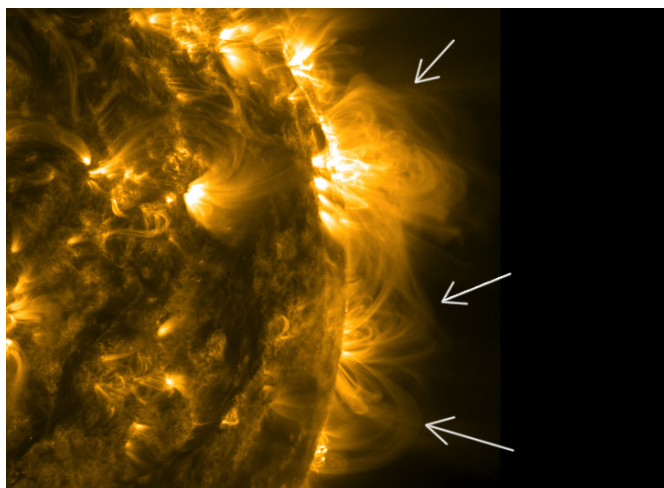
В этом проекте мы с вами будем учиться вручную выделять контуры объектов с изображений и фитировать их на примере колебаний петель в солнечной короне. Для начала разберёмся с тем, что же мы в конечном счёте с вами будем анализировать?

1. Что такое корональные петли?

Корональная петля – это конфигурация плазмы вдоль силовой линии магнитного поля Солнца. Про них пока мало что известно, есть куча моделей структуры петли, профиля интенсивности излучения плазмы и т.д., но пока к единой концепции никто не пришел из-за отсутствия данных, которые бы чётко выделяли одну гипотезу. Мы с вами будем анализировать колебания этих петель, колебания корональных петель – штука, обнаруженная не так давно, их предсказывали ещё в середине 70х, но первые статьи непосредственно о наблюдении их с помощью инструмента TRACE были опубликованы в 1999 году.

Зачем вообще люди их изучают? В идеале, зная параметры колебаний и профиль интенсивности в петле, мы можем локально оценивать магнитное поле в короне Солнца. Оценка магнитного поля на Солнце – достаточно нетривиальная штука, потому что мы не можем непосредственно поместить туда приборы и померить, нам нужно делать это как-то удалённо. Такие методы есть – к примеру измерение величины магнитного поля по эффекту Зеемана (это расщепление уровней атома в магнитном поле, нормально вам это объяснит Барабанов на квантмехе). Но у этого метода есть ограничение по температуре, поэтому мы его не можем применять везде. Поэтому для короны отлично подходит метод оценки по колебаниям петель.

Но какие есть подводные камни? Один из самых больших подводных камней при обработке данных – это плохое разрешение. Для прибора SDO/AIA, данные с которого мы и будем смотреть, 1 пиксел – это 0.6 угловых секунд Солнца. 1 угловая секунда Солнца – это примерно 725 км. Что означает, что объекты, которые меньше этого предела, мы можем детектировать с ооочень маленькой вероятностью, и она, естественно, падает с размерами объекта. Характерный размер корональных петель – от 100 до 500 Мм (М – приставка «мега», т.е. 10^6 метров или тысяч километров). Часто такое низкое разрешение создаёт проблемы в различении набора петель, они выглядят как одно большое пятно:



В принципе, это не сильно мешает анализу в большинстве случаев, так как можно анализировать колебания по достаточно контрастной границе набора петель. Сейчас это, в принципе не так важно. Также плохое разрешение мешает детектированию колебаний с малой амплитудой. Теперь мы подходим к тому, что есть два ~~штука~~ типа колебаний: затухающие и незатухающие.

Затухающие колебания – это как раз те колебания, которые увидели в 1999 году и которые мы с вами будем анализировать. У них достаточно большая амплитуда: в среднем несколько Мм. Что в принципе позволяет увидеть их на видео со спутника невооруженным, так сказать, глазом. Один из механизмов возбуждения таковых колебаний на картинке из статьи моего научрука (*I. V. Zimovets, V. M. Nakariakov, 2014, A&A*):

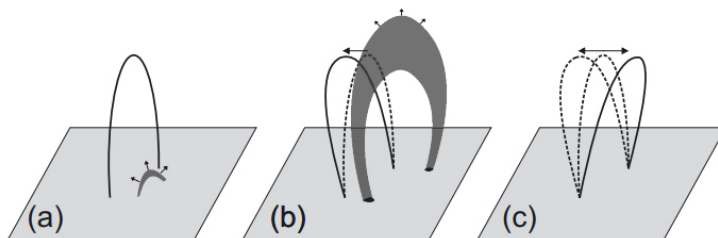
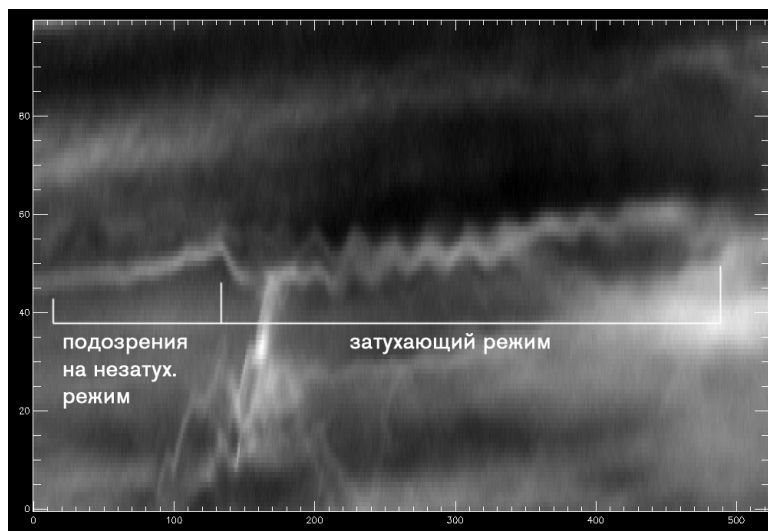


Fig. 2. Schematic illustration of the mechanism for the excitation of kink oscillations of coronal loops, observed in the majority of the studied events. **a)** Pre-eruption state of the active region. **b)** Displacement of a coronal loop (solid black curve) from its equilibrium state (dashed black line) by an erupting and expanding plasma structure, e.g. a flux rope (grey loop-shaped structure). **c)** Oscillatory relaxation of the loop to its equilibrium state after the eruption.

Также есть куча моделей процессов затухания и т.д., в принципе анализ колебаний тоже может в какой-то степени подтверждать и опровергать определённые модели.

Второй режим колебаний: незатухающие. Они уже были детектированы сильно позже из-за своей малой амплитуды: в 2015 году. Просто глазом на видео их увидеть очень сложно просто в силу малости амплитуды (обычно идёт отклонение на 1-2 пиксела). Их заметили на так называемых time-distance maps. Они представляют собой эволюцию интенсивности излучения плазмы в срезе (прямоугольнике 1 на n пикселей) во времени. Пример TD map:



Для изучения такого режима существуют свои методы, их мы касаться не будем. А будем работать только с затухающим режимом.

Часть II

Практическая часть

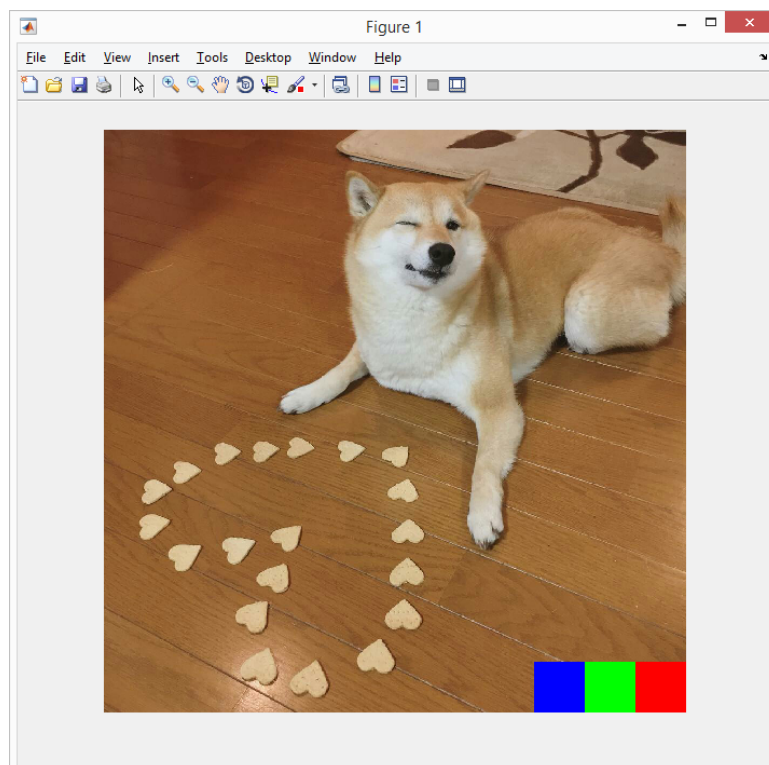
Особой теории по работе с изображениями нет, с теорией по фитам мы с вами разобрались (я надеюсь) во втором задании.

2. Загрузка изображений в МАТЛАВ для работы

Для начала научимся открывать картинки в МАТЛАВ. Качаем из папки проекта файл SOBAKEN.JPG, делаем активной папку, в которую скачали картинку и пишем команду

```
> > imshow sobaken.jpg
```

Получаем открытую картинку собани в окне фигуры:



У функции `imshow` есть полезная опция размера картинки. Например, чтобы картинка вывелась по размеру окна, можно написать:

```
> > imshow ('sobaken.jpg','InitialMagnification','fit')
```

Также вместо `'fit'` можно писать число – это будет процент от реальноно размера картинки. Дальнейший наш шаг – научиться доставать координаты определённого пиксела.

Для этого пишем:

```
> > imshow ('sobaken.jpg', 'InitialMagnification', 40)
[ x,y,P ] = impixel
```

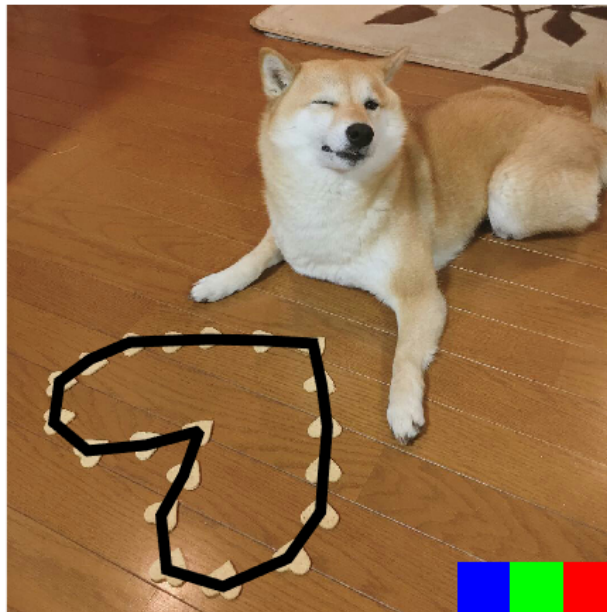
И тыкаем левой кнопкой на красный квадрат, левой кнопкой на зелёный квадрат, правой кнопкой на синий квадрат. В результате получаем три матрицы: матрица иксов для трёх выбранных точек, матрица игреков и матрица, как понятно по значениям, значений RGB каждого выбранного пиксела.

Обращаю ваше внимание на то, что координаты пикселей – это координаты пикселей исходной картинки, а не масштабированной. В этом можно убедиться, кликнув в правый нижний угол картинки и получив что-то около 1080 пикселей, что и есть размер картинки.

Теперь мы, допустим, хотим обвести сердечко на картинке по контуру. Для этого кликаем на каждую печенюшку и не закрываем нашу картинку после того, как нажали, и пишем:

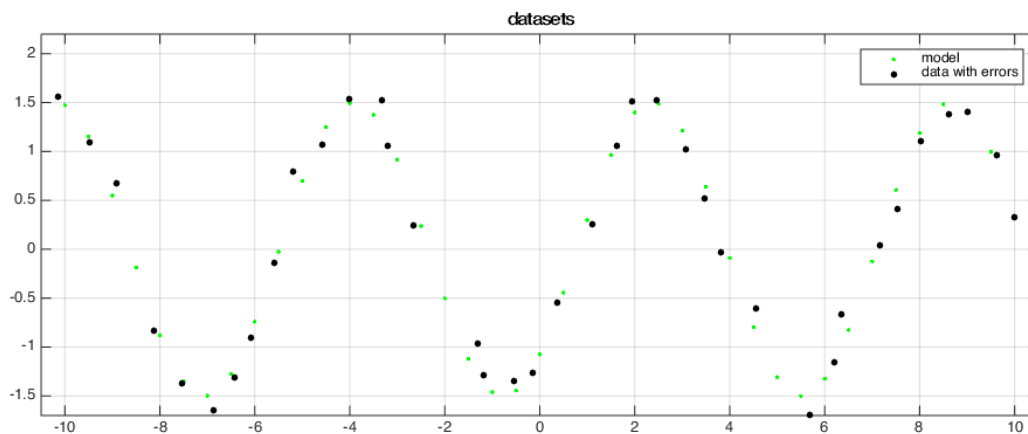
```
> > hold on
plot(x,y,'k','LineWidth',6)
```

Получаем, предсказуемо, это:



3. Фитирование данных

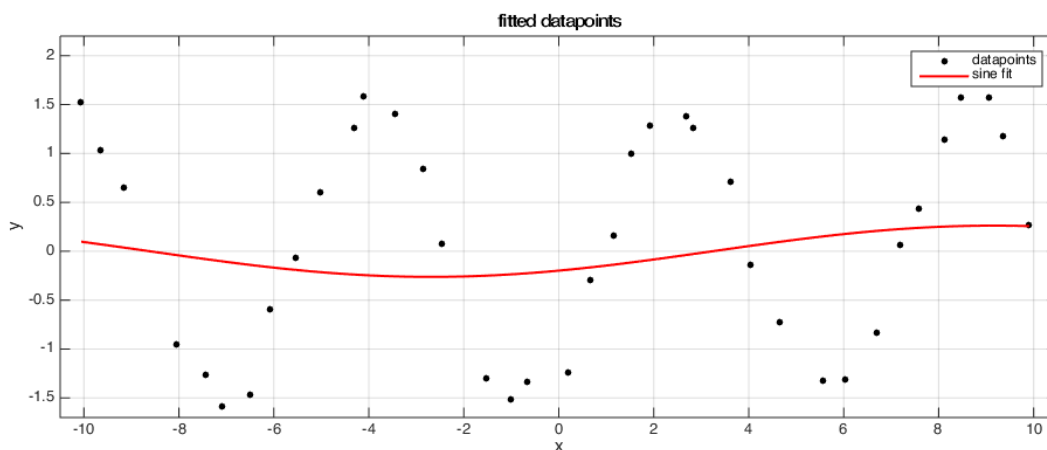
Теперь более пристально посмотрим на задачу об аппроксимации точек какой-либо кастомной функцией. Для этого зададим функцию синуса и добавим для неё шум по обеим координатам, а так же удалим несколько точек совсем:



Теперь для того, чтобы аппроксимировать полученные точки синусом, применим функцию `fitttype`:

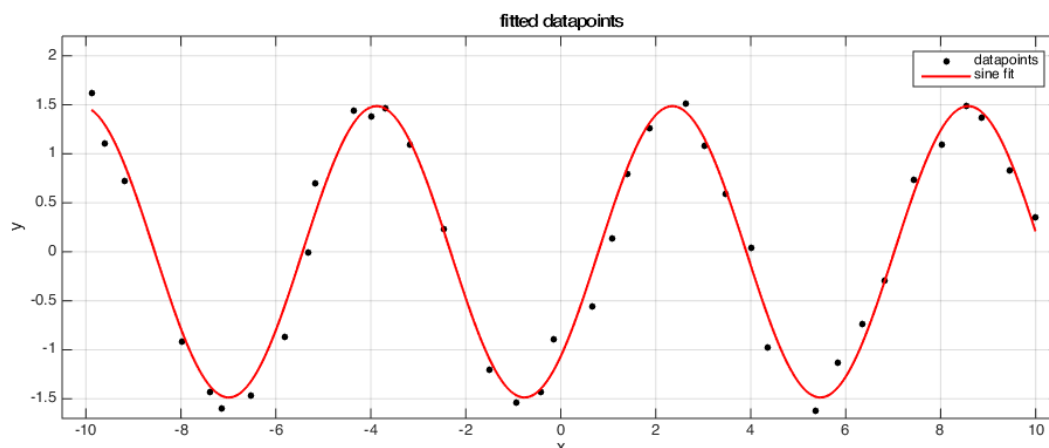
```
ft = fitttype( 'sin1' );
opts = fitoptions( 'Method', 'NonlinearLeastSquares' );
opts.StartPoint = [0 0.1 -0.1];
```

Нижняя строчка задаёт вектор начальных значений для коэффициентов a , b и c модели $y_1 = a \sin(bx_1 + c)$. Мы знаем, что в идеале эти коэффициенты должны равняться 1.5, 1 и -0.8 соответственно. Однако при построении графика нашей аппроксимации мы получим это:



Как же так и что же делать? Ответ достаточно прост: мы выбрали плохие начальные значения нашей аппроксимации. Проблема в том, что наш алгоритм аппроксимации – это **NonlinearLeastSquares**, то есть программа минимизирует квадрат разности модели и данных. А так, как это всё делается численно, наша программа может застрять где-то в локальном минимуме не дойдя до глобального, который нам в нашей модели и нужен. Это достаточно легко представить на пальцах. Представьте поверхность с маленькой ямкой слева и большой справа. Большая ямка – это наши значения, которые близки к модельным. Маленькая ямка – это то, что получилось. Если мы пускаем шарик левее маленькой ямки направо, то он застрянет в маленькой ямке и не достанет до нужной нам большой. Значит надо пускать наш шарик из точки между ямками направо. И тогда он гарантировано попадёт куда надо. Поэтому мы должны выбирать начальную точку как можно ближе к значению, которое нам надо получить. Если поменять начальную точку на `[0`

0.8 -0.5], то мы получим уже то, что надо:



В папке проекта лежит код этой программы. Разбирайтесь. Для выполнения проекта надо так же наугадить, как выводить функциями коэффициенты фита с ошибками.

Часть III

Задачи для самостоятельной работы

Самостоятельно вам нужно будет проанализировать колебания нескольких петель.

Для начала скачайте картинку HOMEWORK.JPG и соедините точки с помощью `plot`. На выходе нужно предоставить картинку в формате `.FIG` и сохранённый в формате `.MAT` массив с координатами точек.

Для трёх событий колебаний нужно оценить длину петли (из предположения, что петля – это половина окружности).

Также оценить параметры колебаний (в величинах минут, Мм [фаза не нужна]). Использовать модель данных: степенной тренд $a_1 x^{b_1}$ и затухающая синусоида $a_2 \sin(b_2 x + c_2) \exp[-d_2 x]$. Посмотреть два способа аппроксимации:

1. аппроксимация степенным трендом, вычитание тренда и аппроксимация затухающей синусоидой;
2. аппроксимация тренд + затухающая синусоида.

Сравнить результаты этих двух способов. С учётом доверительных интервалов, конечно.

Размеры областей изображений:

1. изображение активной области: 1000x1000 внутренней шкалы – 600x600 arcsec;
2. TD maps: одно деление внутренней шкалы по иксу: 12 секунд, по игреку рассчитайте сами из предыдущего пункта.