

## Contents

Markdown . . . . .	1
BibTeX . . . . .	10
JabRef . . . . .	11
Pandoc . . . . .	12
Gnuplot . . . . .	14
Команды для работы с Gnuplot . . . . .	14
Ggobi . . . . .	17
SciDAVis . . . . .	18
Работа с трехмерной графикой . . . . .	20
Анализ графика . . . . .	20
Bazaar . . . . .	22
Cygwin . . . . .	23
Описание . . . . .	23
Несколько основных команд и их аналоги в Windows . . . . .	23
FreeMind . . . . .	25
Возможности программы . . . . .	25

## Markdown

Markdown (маркдаун) — облегчённый язык разметки, созданный с целью написания максимально читаемого и удобного для правки текста, но пригодного для преобразования в языки для продвинутых публикаций

### Элементы разметки

#### Абзацы и переводы строки

Абзац записывается в Markdown весьма просто: это одна или несколько строк текста, отделённые от окружающего текста одной (или более чем одной) пустой строкой. (Пустой строкой считается всякая, которая выглядит пустой; строка, не содержащая ничего, кроме пробелов и символов табуляции, считается пустой.) Обычные параграфы не следует снабжать отступом из пробелов или символов табуляции.

Следствием правила «одна или несколько строк текста» является поддержка Маркдауном таких абзацев текста, в которых имеются «жёсткие переводы

строки» (то есть такие места, в которых автор нажимал на кнопку «ввод», также называемую «Enter»).

Чтобы вставить видимый перенос строки (элемент `<br>`) посредством Markdown, надобно окончить строку двумя (или более) пробелами, и только затем нажать на «ввод» («Enter»).

Безусловно, это требует несколько больших усилий, нежели простое правило «каждый перевод строки преобразуется в `<br>` и становится видимым», однако такое правило для Markdown не подходит. Некоторые (рассматриваемые далее) особенности синтаксиса Markdown — email-подобное блочное цитирование, многоабзачные элементы списков — гораздо лучше выглядят и работают, если их форматировать «жёсткими переводами строки».

## Заголовки

Markdown поддерживает два стиля заголовков: подчёркнутые и выделенные символом «#». Подчёркнутые заголовки подчёркиваются знаками равенства (если заголовок первого уровня) или дефисами (если второго уровня). Годится какое угодно количество подчёркивающих символов «=» или «-». Заголовки, выделенные символом «#», используют от одного до шести таких символов подряд в начале строки; количество символов соответствует уровню заголовка (от первого до шестого). Вот пример:

`# Заголовок первого уровня`

`## Заголовок второго уровня`

`##### Заголовок шестого уровня (H6)`

При желании можно снабжать эти заголовки «закрывающими» символами «#». (Это не обязательно, но вполне возможно, если автор текста полагает, что с «закрывающими» символами «#» заголовок выглядит красивее.) Количество таких конечных символов не обязано соответствовать количеству начальных символов. Уровень заголовка определяется только количеством начальных символов «#»:

`# Заголовок первого уровня #`

`## Заголовок второго уровня ##`

### Заголовок третьего уровня #####

## Цитаты

Markdown использует email-подобный стиль использования символов > для оформления цитат. Поэтому тем авторам, которым привычна электронная почта, уже известен и маркдауновский способ создания цитат. Лучше всего цитата выглядит, если оформить её «жёсткими переводами строки» (см. выше), и каждую строку начать символом «>»:

Это цитата, состоящая из двух абзацев текста. Вы видите, что первый абзац состоит из нескольких строк, каждой из которых предшествует символ закрывающей угловой скобки.

Это второй абзац текста. Обратите внимание на то, что символом цитирования снабжён также и промежуток между абзацами.

Это цитата, состоящая из двух абзацев текста. Вы видите, что первый абзац состоит из нескольких строк, но только первой из них предшествует символ закрывающей угловой скобки.

Это второй абзац текста. Цитаты могут быть вложенными (то есть цитатами внутри цитат), для их разметки используются дополнительные уровни «>»:

Это первый уровень цитирования.

    Это вложенная цитата.

Возвращаемся на первый уровень цитирования. Цитаты могут содержать и другие элементы Markdown — заголовки, списки, кодовые блоки:

1. Это первый элемент нумерованного списка.
2. Это второй элемент нумерованного списка.

Вот пример кода:

## Списки

Markdown позволяет составлять нумерованные и ненумерованные списки.

В качестве маркеров ненумерованного списка используются звёздочки, или плюсы, или дефисы; эти символы могут для этой цели использоваться взаимозаменяемо.

Скажем, вот этот пример:

- Красный
  - Зелёный
  - Синий совершенно равносителен вот этому:
  - Красный
  - Зелёный
  - Синий или вот этому:
  - Красный
  - Зелёный
  - Синий Нумерованные списки используют в качестве маркеров числа с точкою:
1. Чкалов
  2. Байдуков
  3. Беяков Конкретные числа, которые используются при разметке списка, не оказывают никакого влияния на производящийся Маркдауном итоговый HTML-код. По вышеприведённому списку Markdown изготовит такой HTML-код:

Чкалов

Байдуков

Беяков

Если вместо этого записать список так:

1. Чкалов
2. Байдуков
3. Беяков или даже так:
4. Чкалов
5. Байдуков
6. Беяков

Однако в любом случае нумерация нумерованного списка должна начинаться с единицы, хотя создатель Markdown предполагает, что в дальнейшем развитии Markdown может начать поддерживать нумерацию, начинающуюся произвольным числом.

Маркеры списков обычно начинаются с начала строки, но им могут предшествовать до трёх пробелов. За маркером должен следовать либо пробел (один или более), либо символ табуляции.

Для красоты можно снабжать последующие строки списка отступами:

- Это первый элемент списка. Он содержит жёсткие переносы строк, и в начале каждой строки находится отступ, соответствующий началу текста после маркера, расположенного в первой строке.
- Это второй элемент списка. Он оформлен сходным образом: в начале каждой строки его находится отступ, соответствующий началу текста после маркера, расположенного в первой строке. Но если нет надобности, то на это можно не тратить время:
- Это первый элемент списка. Достаточно поставить маркер списка в начале первой строки его, и все остальные строки можно начинать от левого края без отступа.
- Это второй элемент списка. Он оформлен сходным образом: в начале первой строки его находится маркер, а остальные строки следуют за ней без всякого дополнительного отступа. Если элементы списка разделяются пустыми строками, то Markdown обернёт эти элементы тэгами

в своём итоговом HTML-коде. Скажем, вот этот список:

- Святослав
- Владимир будет преобразован к виду

Святослав

Владимир

а зато вот этот:

- Святослав
- Владимир будет преобразован к виду

Святослав

Владимир

Элементы списка могут состоять из нескольких абзацев. Второй, и третий, и каждый последующий абзац элемента должны снабжаться отступом из четырёх пробелов или одного символа табуляции:

1. Это элемент списка, состоящий из двух абзацев. Нетрудно заметить, что оба они снабжены жёсткими переносами строк и отступами, поэтому выглядят красиво.

Это второй абзац первого элемента. Обратите внимание на отступ перед его первой строкою.

2. Это второй элемент списка. Список выглядит красиво, если отступом снабжать каждую строку. Но если нет времени этим заниматься, то для Markdown достаточно снабдить им только первую строку абзаца:
3. Это элемент списка, состоящий из двух абзацев. И хотя не все строки абзацев не снабжаются отступами, Markdown всё же способен понять, что второй абзац является частью списка.

Это второй абзац первого элемента. Обратите внимание, что отступ перед первой строкою абзаца вполне достаточен для Markdown; все остальные строки абзаца внутри элемента списка можно начать и таким образом, как если бы абзац этот не был частью списка.

4. Это второй элемент списка. Если внутри элемента списка располагается цитата, то надобно снабжать отступом разделители «<>», цитате предшествующие:

- Этот элемент списка содержит цитату:

Вот эта цитата, содержащаяся внутри элемента списка.  
Чтобы поместить кодовый блок внутри элемента списка, этот кодовый блок надобно снабдить двукратным отступом, то есть либо восемью пробелами, либо двумя символами табуляции:

- Этот элемент списка содержит кодовый блок:

```
первая строка кода;  
вторая строка кода;  
...
```

## Блоки кода

Блоки отформатированного кода приводятся в статьях о программировании или о разметке текста, когда возникает надобность процитировать исходники. В отличие от обычных абзацев, переносы строк в кодовом блоке воспринимаются буквально. Markdown обрамляет кодовый блок сразу двумя элементами HTML — и

, и .

Чтобы создать блок кода в Markdown, достаточно просто снабдить каждую строку блока отступом, состоящим из четырёх пробелов или одного символа табуляции. Например, вот этот текст:

Это обычный абзац.

А это кодовый блок .

будет преобразован Маркдауном в следующий HTML-код:

Это обычный абзац.

Один уровень отступа (4 пробела или 1 символ табуляции) устраняются из каждой строки кодового блока. Например, вот этот текст:

Это пример на языке AppleScript:

```
tell application "Foo"
    beep
end tell
```

будет преобразован к виду

Это пример на языке AppleScript:

Блок кода продолжается до тех пор, покуда не отыщется строка без отступа (или конец текста).

Внутри кодового блока амперсанды («&») и угловые скобки («<» и «>») автоматически преобразуются в HTML-сущности (HTML entities). Поэтому нетрудно использовать Markdown для включения примеров на языке HTML: достаточно вставить их и снабдить отступом, а Markdown позаботится о кодировании амперсандов и угловых скобок. Скажем, вот этот пример:

```
<div class="footer">
```

```
&copy; 2004 Foo Corporation  
</div>
```

примет вид следующего HTML-кода:

```
<code>&lt;div class="footer"&gt;  
    &amp;copy; 2004 Foo Corporation  
&lt;/div&gt;  
</code>
```

Обычный синтаксис Markdown не обрабатывается внутри блоков кода. Например, звёздочки внутри кодового блока считаются просто звёздочками в буквальном смысле. Так что нетрудно использовать Markdown, чтобы на нём писать о его собственном синтаксисе, приводя наглядные примеры.

**Горизонтальная черта** Чтобы создать горизонтальную черту (соответствующую HTML-элементу

), достаточно поместить три (или более) дефиса, или звёздочки, или символа подчёркивания, на отдельной строке текста. Если угодно, между дефисами или звёздочками можно располагать пробелы. Каждая из следующих строк соответствует горизонтальной черте:

```
-----  
-----  
-----  
-----  
-----
```

## Гиперссылки

Markdown поддерживает три стиля оформления гиперссылок:

с немедленным указанием адреса; подобные снóскам, простую вставку URL. Первая пара стилей предполагает, что помимо URLа существует ещё текст ссылки; в разметке он [заключается в квадратные скобки].

**Гиперссылка с немедленным указанием адреса**

Чтобы создать ссылку с немедленным указанием адреса, надобно немедленно после закрывающей квадратной скобки поместить обычные (круглые) скобки; в



этих круглых скобках приводится тот URL, на который указывает гиперссылка, а за ним может ещё быть указан (в кавычках) всплывающий заголовок-подсказка ссылки. Вот пример:

Это пример ссылки с немедленным указанием адреса. Эта ссылка снабжена всплывающим заголовком-подсказкою.

А для этой ссылки заголовок не указан. Из этого примера выйдет следующий HTML-код:

Это пример ссылки с немедленным указанием адреса. Эта ссылка снабжена всплывающим заголовком-подсказкою.

А для этой ссылки заголовок не указан.

Для ссылок на локальный ресурс (то есть расположенный на том же сервере) можно использовать относительные пути:

Подробности приводятся на странице About. Гиперссылка, подобная сноске

Ссылка, подобная сноске, вместо целевого адреса использует вторую пару квадратных скобок, внутри которых помещается метка, идентификатор ссылки:

Это пример ссылки, подобной сноске. Для разделения двух пар квадратных скобок может использоваться необязательный пробел:

Это пример ссылки, подобной сноске. Затем, где угодно в документе, следует определить эту метку ссылки, для чего используется отдельная строка следующего примерно вида:

Строка эта состоит из следующих элементов:

Идентификатор ссылки, окружённый квадратными скобками (которым может предшествовать необязательный отступ — от одного до трёх пробелов). Двосточие, Один или несколько пробелов (или символов табуляции). URL гиперссылки. Необязательный заголовок (всплывающая подсказка) гиперссылки, заключённый либо в двойные или одиночные кавычки, либо в скобки. Три следующие определения ссылки совершенно равносильны:

```
[foo]: http://example.com/  "А здесь необязательный заголовок"  
[foo]: http://example.com/  'А здесь необязательный заголовок'  
[foo]: http://example.com/  (А здесь необязательный заголовок)
```

URL гиперссылки может (но не обязательно) быть помещён в угловые кавычки:

Можно поместить заголовок на следующей строке, а перед ним отступ из пробелов (или символов табуляции); с длинными URL такая запись выглядит красивее:

Такие определения ссылок используются только для создания самих ссылок при обработке текста в Markdown; из итогового HTML-кода определения ссылок убираются.

Идентификаторы ссылок могут состоять из букв, цифр, пробелов и знаков пунктуации, однако они не чувствительны к регистру. Например, вот эти две ссылки: `[link text][a]` `[link text][A]`

`[link text][a]` `[link text][A]` совершенно равносильны.

## BibTeX

BibTeX — программное обеспечение для создания форматированных списков библиографии. BibTeX используется совместно с LaTeX'ом и входит во все известные дистрибутивы TeX и LaTeX.

BibTeX был создан Ореном Паташником и Лесли Лэмпортом в 1985 году. BibTeX позволяет легко работать со списками источников, отделяя библиографическую информацию от её представления. Принцип отделения содержимого от его представления использован как в самом LaTeX'е, так и в XHTML, CSS и др.

- список литературы генерируется автоматически по всем ссылкам `[?, ?]` упомянутым в тексте;
- можно использовать единую библиографическую базу (bibфайл) во всех своих текстах, во всех работах отдела, и т. д.;
- легко обмениваться библиографическими базами с коллегами;
- нет необходимости помнить правила оформления библиографии, так как BibTeX делает эту работу автоматически с помощью стилевых bst-файлов. `[@wiki]` `[@bibtex]`

BibTeX использует bib-файлы специального текстового формата для хранения списков библиографических записей. Каждая запись описывает ровно одну публикацию — статью, книгу, диссертацию, и т. д.

Bib-файлы можно использовать для хранения библиографических баз данных.

Многие программы, работающие с библиографиями, (такие, как JabRef) могут экспортировать ссылки в bib-формат.

Каждая запись выглядит следующим образом:

```
@ARTICLE{tag,  
  author = {Список авторов},  
  title = {Название статьи},  
  year = {год},  
  journal = {Название журнала}  
}
```

BibTeX использует bst-файлы для описания того, как bib-записи преобразуются в текст на LaTeX'е. Каждый bst-файл представляет собой программу на простом стековом языке программирования, напоминающем Forth или PostScript. Есть программы, позволяющие генерировать .bst-файлы автоматически (например, custom-bib или Bib-it).

## JabRef

JabRef — это система управления библиографической информацией, которая использует BibTeX, как нативный формат. JabRef предоставляет удобный интерфейс для редактирования файлов BibTeX, импортирования данных из онлайн научных баз данных и для поиска и управления BibTeX файлами. Приложение написано на языке программирования Java, и является кроссплатформенным.

Возможности JabRef

- Полностью совместим с BibTeX
- Полнотекстовый поиск по всей библиографии.
- Импорт различных форматов: BibTeXML, CSA, Refer/Endnote, Web of Knowledge, SilverPlatter, Medline/Pubmed (xml), Scifinder, OVID, INSPEC, Biblioscape, Sixpack, JStor and RIS.
- Экспорт в разных форматах HTML, Docbook, BibTeXML, MODS, RTF, Refer/Endnote и OpenOffice.org.
- Группировка по любым полям BibTeX, ключевым словам.

- Интеграция с десктоп-окружением: запуск программ просмотра PDF/PS, браузера, вставка цитирований в LyX, Kile, LatexEEditor, Emacs, Vim и WinEdt, OpenOffice.org (с помощью плагина)
- Поддержка плагинов — расширений.
- Автоматическое создание BibTeX ключей

Интерфейс программы легко описывается по верхнему меню окна программы:

1. File - группа для создания или открытия базы данных, bib-файла, коллекции bib-файлов, синхронизации БД с внешним источником, импорта и экспорта БД или ее частей. Также присутствует возможность сохранять различные сессии для работы разных пользователей. 2. Edit - стандартная группа команд для редактирования и подсветки результатов. 3. Search - поиск элементов, а также, обнаружение дубликатов и экспорт в буфер обмена команд на цитирование. 4. View - отображение таблиц и шрифтов. 5. BibTeX - создание новой библиографической записи с выбором ее типа, редактирование ее дополнительных параметров. 6. Tools - связь с внешними источниками или программами, экспортирование в редактор WinEdt, открытие привязанного текста статьи из pdf-файла или URL ссылки или DOI, автогенерация ключевых слов. 7. Plugins - подключение дополнительных программных модулей. 8. Options - свойства программы, установки по умолчанию, настройки экспорта. 9. Help - меню помощи и указатель на сайт разработчика. Сортировка по полю происходит нажатием на имя столбца. Для редактирования записи дважды кликните на ней, после этого перемещайтесь между горизонтальными вкладками с группами обязательных и необязательных полей. Последняя вкладка отвечает за содержимое bib-файла, которое можно редактировать вручную.

## **Pandoc**

Pandoc — универсальная утилита («швейцарский нож») для работы с текстовыми форматами. Основная сфера применения — форматирование математических и технических текстов.

Pandoc представляет собой кроссплатформенную программу с командным интерфейсом, способную преобразовывать тексты в самых разнообразных разметках в многочисленные выходные форматы.

Так, например с использованием pandoc можно конвертировать исходные документы в разметках ASCII doc, Wiki, Markdown в HTML. Если установить LaTeX, то становится возможным получение и PDF.

Действительно, pandoc справляется с конвертированием без каких-либо потерь информации. При конвертировании из формата Markdown поддерживается чтение трех параметров метаданных — заголовка, автора и даты документа. Поддерживается так же передача параметров командной строки для установки некоторых специфических свойств, например языка документа. Есть возможность задать свой шаблон выходного документа, до некоторой степени видоизменяя его.

### **Основные команды**

Ниже рассмотрим самые основные параметры и опции Pandoc, В остальных случаях желательно познакомиться с довольно объёмным руководством на сайте или через руководство man.

Pandoc - программа консольная, и ей требуется передавать в качестве параметров имя входного файла, а и при помощи опции “-o” - целевого файла. Программа может понимать входной и выходной формат разметки по расширению файла.

```
pandoc input.md -o output.html
```

Однако можно указать входной и выходной форматы при помощи специальных опций. Выходной формат задаётся опциями “-t” или “-to”, а входной - опциями “-f” или “-from”. Например, следующая команда перекодирует файл input.md с разметкой Markdown в файл output.txt с html-разметкой

```
pandoc input.md -o output.txt -t html
```

Если опции форматов или расширения файлов не указаны или не известны для Pandoc, то по умолчанию программа считает форматом входного файла Markdown, а выходного - HTML.

### **Математика**

Интересной особенностью Pandoc является поддержка конвертирования математических формул из разметки LaTeX в представление HTML. Для вывода математики в HTML используются на выбор несколько специальных математических движков на основе MathML, Java-Script, онлайн-сервисов, код которых будет вставлен в сконвертированный HTML-файл.

Для конвертирования математических формул можно использовать следующие опции:

`--mathml` - преобразует формулы LaTeX в разметку MathML;

`--webtex` - преобразует формулы LaTeX при помощи онлайн-сервиса Google Chart API;

`--mathjax` - преобразует формулы LaTeX при помощи расширения MathJax для MediaWiki;

`--latexmathml` - преобразует формулы LaTeX при помощи JS-библиотеки Latexmathml.

## Gnuplot

**Gnuplot** — свободная программа для создания двух- и трёхмерных графиков. Gnuplot имеет собственную систему команд, может работать интерактивно (в режиме командной строки) и выполнять скрипты, читаемые из файлов. Также используется в качестве системы вывода изображений в различных математических пакетах: GNU Octave, Maxima, Reduce и других.

Gnuplot выводит графики как непосредственно на экран (интерактивный режим), так и в файлы различных графических форматов (командный режим работы), таких как PNG, EPS, SVG, JPEG и множество других. Программа также может генерировать код на LaTeX, позволяя использовать шрифты и формулы LaTeX.

Основными достоинствами gnuplot являются: - активная поддержка, - стабильная работа, - бесплатность и открытый исходный код, - кроссплатформенность (Linux, Windows, MacOS), - интерактивный ввод или выполнение скриптов, - богатый набор встроенных функций, - множество примеров, - гибкий и эффективный язык программирования, - поддержка вывода в множество графических форматов и устройств.

### Команды для работы с Gnuplot

- `cd` Изменение текущего каталога (эта команда очень полезна при визуализации числовых данных из файлов).

- `pwd` Команда имеет тот же смысл, что и аналогичная команда в Linux — она позволяет посмотреть, какой каталог в текущий момент активен
- `reset` Позволяет убрать все изменения в настройках программы, сделанные пользователем.
- `help` Можно сказать, что это одна из наиболее часто используемых команд. Её назначение — вызов справки по Gnuplot. Если вызвать команду без аргументов, то вызывается общий файл справки. При использовании аргумента (им является любая команда Gnuplot) происходит обращение к разделу справки, отвечающего аргументу.

Кроме команд, требуется знать ещё несколько параметров Gnuplot

- `xrange` Параметр устанавливает диапазон независимой переменной, в котором будет рисоваться график. Параметр можно изменять с помощью команды `set xrange[{min}:{max}]`. `{min}` и `{max}` — необязательные параметры, определяющие нижнюю и верхнюю границы интервала. В качестве значений для них может выступать число или же символ `*` (в этом случае соответствующая граница будет вычисляться автоматически). Примеры:
  - `set xrange[0:10]` Нижняя граница 0, верхняя — 10
  - `set xrange[*:1.2]` Теперь нижняя граница вычисляется автоматически, а верхняя устанавливается равной 1, 2
  - `set xrange[-0.98:*)` Нижняя граница равна -0, 98, верхняя вычисляется
 Gnuplot “Откатить” сделанные изменения можно с помощью `reset -yrange` Параметр делает точно то же самое, что и `xrange`, только в отношении оси ординат.
- `key` Отвечает за легенду рисунка, то есть метку, располагающуюся в верхнем правом углу графика. Лично мне нравятся графики без какой-либо избыточной информации, поэтому чаще всего приходится использовать команду `set nokey`, отменяющую появление легенды. Однако можно и принудительно установить легенду с помощью команды `set key 'text'`. Пример: `set key 'This is the graph of Bessel function'`
- `terminal` Этот параметр определяет, куда именно Gnuplot будет выводить построенный график. По умолчанию это значение устанавливается таким образом, что пользователь получает график на мониторе. Если же требуется получить “твёрдую” копию (например, напечатанный график или же графический файл), то значение надо переустановить с помощью `set terminal {terminal-type}`. Возможные значения `{terminal-type}` можно

посмотреть с помощью `help set terminal`. Примеры: • `set terminal postscript` После Gnuplot будет выводить данные в формате PostScript. • `set terminal X11` X11 — это имя дисплея в операционной системе Linux, так что после этой команды Gnuplot будет выводить графики на монитор. • `set terminal windows` То же самое, что и в предыдущем случае, но только в операционных системах Windows • `set terminal latex` После такой команды график будет печататься в графике LATEX. Если Вы пользуетесь LATEX, то это неплохой способ для вставки графиков в текст, однако надо иметь в виду, что размеры получающихся файлов могут оказаться большими.

- `output` Имя файла или устройства для вывода данных. Команда для установки `set output { 'filename' }` Если Вы не установите его значения, но попытаете изменить текущее значение `terminal`, то очередной график Gnuplot напечатает прямо на экране (в текстовом виде). Например, после выполнения таких команд `gnuplot>set terminal postscript gnuplot>plot sin(x)`

Пример: `set output "myfile.ps"` После этого у Вас возникает на диске (в текущем каталоге) пустой файл с именем `myfile.ps`. После выполнения рисования в файл запишется содержимое — график в выбранном Вами формате. Важное замечание: если требуется получить несколько графиков, причём каждый из них должен быть записан в свой файл, то `set output` надо выполнять перед каждым процессом рисования, иначе все графики будут записаны в один и тот же файл.

### Примеры работы с Gnuplot

Для построения графика в режиме консоли необходимо набрать

```
gnuplot> plot cos(x)
```

команда

```
gnuplot> plot "xyz.dat" use 1:2 with lines
```

Вызовет создание окна с графиком зависимости, данные для которой берутся из 1 колонки файла `xyz.dat` для X-оси и 2 колонки - для Y-оси. Параметр `with lines` (можно просто `w l`) указывает, что график нужно строить в виде линии.

Чтобы построить одновременно несколько графиков из данных, записанных в колонках одного файла, нужно несколько раз считать эти данные из файла в одной и той же команде `plot`



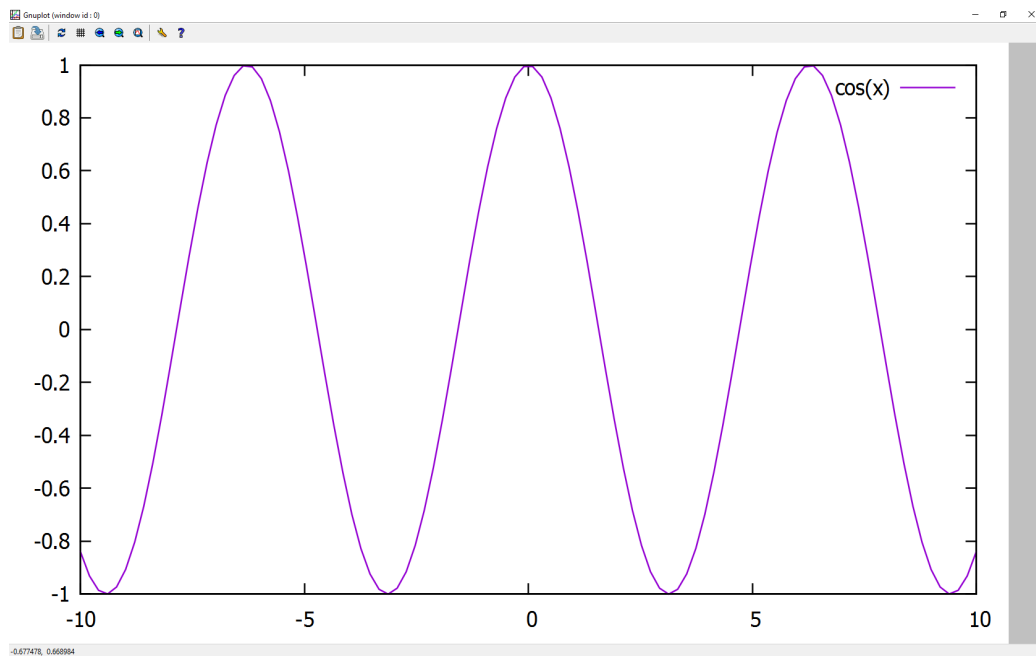


Figure 1: График косинуса

```
plot "xyz.dat" u 1:2 , "xyz.dat" u 1:3 , "xyz.dat" u 2:3
```

## Ggobi

GGobi — это программа для наглядного представления многомерных данных. Она может создавать как высокодинамичные и интерактивные графики, например экскурсии (вращение графика в более чем трех измерениях), так и привычные диаграммы рассеивания, гистограммы и построения с параллельными координатами. Г связаны закраской и идентификацией. В данной программе удобно по графику анализировать данные, выводить закономерности. Работает с файлами форматов: csv, xml.

Основные преимущества Ggobi

- Хорошее средство для работы с 3d графикой
- Создание графики высокого качества для различных видов публикаций.
- Отличное средство для кластерного анализа
- Работа с данными из файлов в формате CSV, XML

- Адаптировано для работы с R



Figure 2: Пример кластерного анализа

## SciDAVis

SciDAVis — свободное бесплатное программное обеспечение для анализа научных данных, их визуализации и подготовки к печати, распространяемое под лицензией GNU GPL. SciDAVis может строить различные типы 2D и 3D-графиков (линейные графики, точечные графики, трёхмерные гистограммы, объёмные круговые гистограммы, трёхмерные поверхности) на основе данных, импортированных из ASCII файлов, введённых вручную или вычисленных по формулам[2]. Данные содержатся в отдельных таблицах (англ. spreadsheets), содержащих данные в столбцах (значения по осям X и Y при построении 2D-графиков) или в виде матриц (при построении 3D-графиков). Таблицы, графики и сопутствующие примечания собираются в проекты и могут быть организованы с использованием директорий.

Встроенные средства анализа SciDAVis включают в себя построчную/постолбцовую статистику таблиц, свёртку/обратную свёртку, быстрый фурье-анализ и

основанные на быстром фурье-анализе фильтры данных. Аппроксимация данных может производиться как с помощью функций, заданными пользователем, так и с помощью встроенного набора линейных и нелинейных функций аппроксимации, включая мультипиковую аппроксимацию (англ. multi-peak fitting) с использованием библиотеки GNU Scientific Library.

Графики могут быть сохранены в нескольких растровых графических форматах файлов, а также в форматах PDF, EPS или SVG. Поддерживается встроенное вычисление математических выражений и имеется опциональный скриптовый интерфейс для языка Python. Графический интерфейс пользователя создан с использованием библиотеки Qt.

SciDAVis сочетает простой в освоении и интуитивно понятный интерфейс с обширными возможностями, такими, как возможность написания скриптов и создания собственных модулей.

Импорт данных из других программ, который осуществляется командой Файл/Импорт ASCII. Программа предоставляет несколько вариантов импорта данных:

- Текстовый файл (.TXT), в котором числа записаны по столбцам. При открытии текстовых файлов в SciDAVis есть возможность указать разделитель между столбцами, разделитель дробной части и символ, который отделяет в больших числах тысячи, миллионы и т.д. для придания данным «удобочитаемого» вида.
- Текстовый файл, в котором числа разделяются запятой (SCV).
- Файл данных в формате .DAT.
- Вставка из буфера обмена.

### **Создание графика по данным таблиц**

При таком варианте для построения графика следует произвести такие действия:

Выделить область данных в столбце таблицы с названием координаты [Y]. При этом можно использовать не все данные столбца, а только его определенную часть.

Создать график. Это можно сделать несколькими вариантами:

- вызывать контекстное меню в области столбца с результирующим признаком [Y],
- выбрать из него последовательно пункты График и тип графика;
- еще быстрее можно выполнить это действие можно из пункта главного меню График;
- при наличии панели инструментов Plot (график) щелкнуть кнопку с изображением нужного типа графика.

Построенный график отображается в новом окне. Он содержит несколько элементов: общий заголовок, заголовки осей, масштабные координаты и графический образ. Легенда обычно располагается в верхней левой части графика. Она отображает по данным какой таблицы построен график и каким цветом отображаются данные.

### **Работа с трехмерной графикой**

Типы графиков, которые позволяет строить система

- Поверхность в виде ленты.
- Столбики, положение которых на плоскости задается двумя координатами, а третья координата определяет их высоту.
- «Пузырьковая диаграмма» (Scatter Plot).
- Трассектория.

### **Анализ графика**

Система предоставляет пользователю ряд аналитических возможностей. Их рассмотрение представляет тему отдельной статьи, поэтому ограничусь просто их перечислением и самой краткой аннотацией.

Вся аналитика содержится в пункте главного меню Анализ. К аналитическим возможностям программы относится следующее:

- Транслировать. Перемещение графика по вертикали или горизонтали. Осуществляется так:
- Выполнить команду Анализ/Транслировать/Вертикально (Горизонтально).

- Применить инструмент считывания данных и осуществить двойной щелчок на линии графика.
- Осуществить двойной щелчок на вертикальной (или горизонтальной линии). В результате этого система рассчитает разницу между двумя значениями на оси  $x$  или  $y$  и добавит ее ко всем точкам графика, что повлечет перемещение графика на плоскости. Одновременно изменятся данные в таблице.
- Дифференцирование. После выполнения команды Анализ/Дифференцирование... (и выбора графика, если на плоскости их отображено несколько) в новом окне появляется график производной. Одновременно для рассчитанных данных создается новая таблица.
- Интегрирование. Для интегрирования от базовой нулевой линии в программе применяют метод трапеций. После выполнения команды Анализ/Интегрирование... появится окно, в котором будет предложено ввести порядок, количество итераций, допуск, верхний и нижний предел. Результаты интегрирования, в частности пиковые значения  $x$  и  $y$ , площадь отображаются в специальном окне «Журнал результатов».
- Сглаживание данных (Smoothing). Сглаживание является специальной операцией усреднения. Для сглаженных данных создается скрытая таблица, имеющая тип «скрытый». Программа имеет три варианта сглаживания данных, доступные после выполнения команды Анализ/Сглаживание...:
- Сглаживание, использующее фильтр Савицкого-Голая. В окне «Опции сглаживания» задается степень полинома для сглаживания (максимум «9»). Именно этот параметр позволяет улучшать приглаживание.
- Скользящее среднее. Здесь необходим комментарий относительно названия варианта сглаживания. Имеет место ошибка в переводе. Это не метод скользящих средних, а метод увеличения периодов, суть которого заключается в том, что первичные данные заменяются средними по интервалам. В окне «Опции сглаживания» пользователь задает только количество точек в интервале, по которым рассчитывается средняя. Принято применять нечетное количество интервалов, как правило, «3» или «5».
- БПФ-фильтр (Быстрое преобразование Фурье), FFT (быстрое превращение Фурье). В окне «Опции сглаживания» определяют только число точек

для сглаживания.

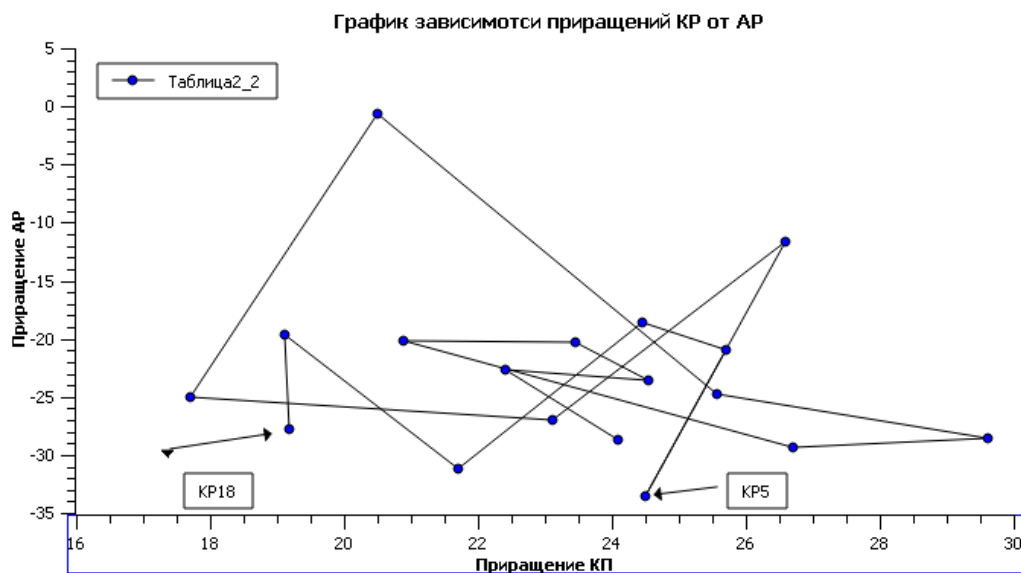


Figure 3: Приращение в SciDAVis

## Bazaar

Bazaar (ранее известная как Bazaar-NG, имя утилиты командной строки bzr) — распределённая система управления версиями, разработка которой спонсируется фирмой Canonical Ltd.. Система Bazaar разработана с целью облегчить работу над развитием свободных и открытых проектов для всех желающих.

Команда разработчиков фокусируется на том, чтобы сделать систему лёгкой в использовании, но при этом точной в деталях и очень гибкой, подстраиваемой под конкретные нужды пользователей. Также большое внимание уделяется вопросам работы со множеством веток и их последующим объединением. Bazaar может использоваться как одним разработчиком для работы над множеством локальных веток, так и группой разработчиков, совместно работающих в сети.

Система контроля версий Bazaar написана на языке программирования Python. Существуют установочные пакеты для основных дистрибутивов

Linux, инсталляторы для Mac OS X и MS Windows. Bazaar — это свободное программное обеспечение, в настоящее время является частью проекта GNU.

## **Cygwin**

Cygwin это UNIX-подобная среда и интерфейс командной строки для Microsoft Windows. Cygwin обеспечивает тесную интеграцию приложений, данных и ресурсов Windows с приложениями, данными и ресурсами UNIX-подобной среды. Из среды Cygwin можно запускать обычные приложения Windows, также можно использовать инструменты Cygwin из Windows

### **Описание**

Cygwin представляет собой инструмент для портирования ПО UNIX в Windows и представляет собой библиотеку, которая реализует интерфейс прикладного программирования POSIX на основе системных вызовов Win32. Кроме того, Cygwin включает в себя инструменты разработки GNU для выполнения основных задач программирования, а также и некоторые прикладные программы, эквивалентные базовым программам UNIX. В 2001 году в Cygwin был включён пакет X Window System.

Кроме того, Cygwin содержит библиотеку MinGW, позволяющую работать с библиотекой Microsoft MSVCRT (Windows API); библиотека MinGW менее требовательна к объёму оперативной и дисковой памяти, распространяется под более свободной лицензией и может работать с любым программным обеспечением, но функциональные возможности спецификации POSIX реализованы в ней не так полно, как в Cygwin.

Red Hat реализует библиотеку Cygwin под лицензией GNU GPL, таким образом, её можно использовать только в программах, распространяемых под GPL. Для распространения программ, использующих библиотеку Cygwin под лицензией, отличной от GPL, необходимо приобретение лицензии у RedHat.

### **Несколько основных команд и их аналоги в Windows**

`ls = dir` — посмотреть содержание директории;

```
cd = chdir — перейти в другую директорию;  
cp = copy — скопировать файл;  
mv = mv — переместить/переименовать файл;  
rm = del — удалить файл;  
mkdir = mkdir — создать директорию;  
pwd — вывести текущую директорию.
```

Одна из удобнейших особенностей Bash — это автодополнение (autocompletion).  
Создадим командой cat несколько файлов:

```
cat > first_file ENTER
```

введите что-то, затем Ctrl-C,

и то же самое для другого файла:

```
cat > second_file ENTER.
```

Убедитесь, что файлы существуют:

```
ls
```

Теперь наберите любую команду, например, ls, и первую букву первого файла. Нажмите TAB. Bash сам допишет имя файла. Если файлы начинаются одинаково, например, file\_one и file\_two, Bash допишет только file\_ и подождет, пока Вы введете еще одну букву, чтобы устранить многозначность.

## ПРИМЕРЫ

Несколько примеров использования некоторых утилит.

Допустим, у Вас есть текстовый файл in.txt. Посчитать количество слов в нем Вы можете так:

```
wc -w in.txt
```

Количество строк:

```
wc -l in.txt
```

Найти строки, содержащие последовательность символов Larry:

```
grep 'Larry' in.txt
```

Разбить строки на слова, заменяя пробелы на символы конца строки:

```
cat in.txt | gawk 'gsub(" +", "\n")'
```



Команда

```
gawk 'gsub(" +","\\n")'
```

Сделать то же самое, но с возможностью удобного просмотра (PageUp, PageDown, выйти — q)

```
cat in.txt | gawk 'gsub(" +","\\n")' | less
```

Отсортировать строки (слова):

```
cat in.txt | gawk 'gsub(" +","\\n")' | sort | less
```

Убрать одинаковые строки (слова):

```
cat in.txt | gawk 'gsub(" +","\\n")' | sort | uniq | less
```

Подсчитать количество уникальных строк (слов):

```
cat in.txt | gawk 'gsub(" +","\\n")' | sort | uniq | wc -l
```

Убрать одинаковые строки (слова), считая их количество:

```
cat in.txt | gawk 'gsub(" +","\\n")' | sort | uniq --count | less
```

## FreeMind

FreeMind — свободная бесплатная программа для создания диаграмм связей (правильное название понятия, чаще известного как «карты памяти», «mind maps»). FreeMind написана на Java и распространяется согласно GNU General Public License. Программа обладает расширенными возможностями экспортирования. Экспорт XHTML позволяет создать карту-схему с разветвленной структурой и ссылками на внешние источники. Наглядность представления информации. Поддержка импорта и экспорта в форматы: PNG, JPEG, XML, HTML, XHTML, OpenDocument Text; плагин для экспорта в SVG и PDF.

### Возможности программы

- Поддержка вкладок — вы можете одновременно работать с несколькими открытыми картами, легко переключаясь между ними.

- различные стили форматирования текста и узлов вашей mindmap.
- легкая в освоении (три клавиши — Insert, F2 и Delete — вам помогут сразу начать с ней работать)
- Около 30 иконок, для улучшения восприятия информации.
- Наглядность полученных карт.
- Скрытие ветвей.
- Возможность использования HTML для форматирования узлов.
- Графическое связывание узлов.
- Ссылки на другие карты памяти, веб-страницы и внешние файлы.
- Поиск по отдельным ветвям.
- Импорт и экспорт списков.
- Шифрование: возможно шифрование как документа в целом, так и создание отдельных шифрованных узловых элементов диаграммы (для доступа к дочерним узлам потребуется ввод пароля).