

# Руководство пользователя программой BaseOfModelling

Программное обеспечение, представленное в данном руководстве пользователя, предназначено для использования на практических занятиях курса «Основы моделирования».

Программа **BaseOfModelling** разработана в среде Matlab 7.0.1 (может быть использована и в более поздних версиях, также в студенческой).

Программа **BaseOfModelling** содержит следующие подпрограммы, доступные из меню **Модель**:

1. `sys` — для моделирования динамических систем (непрерывных и дискретных во времени) и представления результатов в виде графиков или фазовых диаграмм;
2. `gradsys` — для моделирования непрерывных во времени динамических систем, которые представлены градиентом;
3. `bif` — для моделирования диаграмм раздвоения;
4. `fractal→complex` — собрание подпрограмм, иллюстрирующих сложные динамические системы;
5. `fractal→IFS` — построение геометрических фракталов на основе систем итерируемых функций.

Каждый из наборов подпрограмм находится в одноименных каталогах, содержит демонстрационный набор примеров-вариантов заданий.

## Настройка программы

Для работы программы **BaseOfModelling** необходимо среди путей поиска прописать каталог **Work** (в нем она находится) с его подкаталогами.

Добавление каталога **Work** в пути поиска осуществляется при выборе пункта **Set Paths** меню **File** рабочей среды Matlab. Нажмите кнопку **Add Folders with Subfolders** и в появившемся диалоговом окне выберите каталог **Work**. Для сохранения изменений для последующих запусков программы нажмите **Save**.

Для запуска программы из рабочей среды Matlab необходимо ввести команду **BaseOfModelling** (рабочим каталогом должен быть тот, в котором находится эта программа).

После запуска программы на экране раскроется окно (рис. 1) Полоса

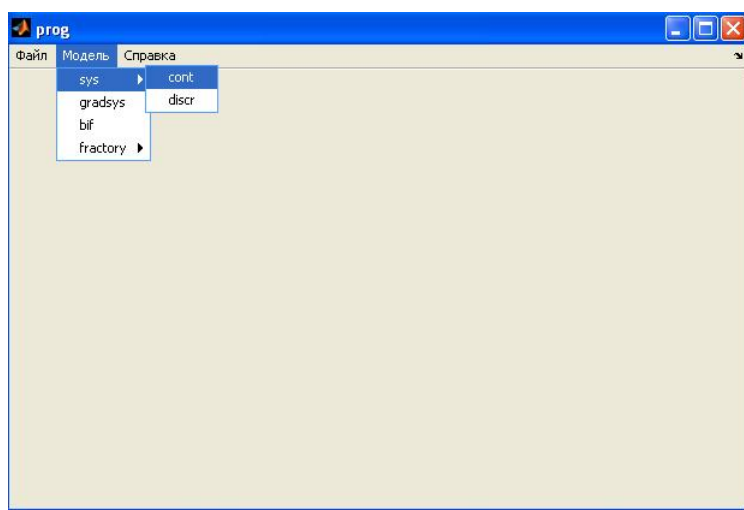


Рис. 1: Окно главной формы программы **BaseOfModelling**

меню является основным доступом ко всем командам меню. Если полоса меню активна, то заголовок меню будет высвечен; это текущее выбранное меню. Если за командой следует ►, то команда ведет в другое меню. Команда без стрелки указывает, что как только вы ее выберете, откроется соответствующее окно, в котором можно моделировать.

ПРИМЕЧАНИЕ: Символы кириллицы могут неправильно отображать-

ся в нелокализованной версии Matlab. Один из способов заключается в изменении текстового файла `matlabrc.m`, находящегося в подкаталоге `toolboxes\local` основного каталога Matlab. Используя любой текстовый редактор, добавьте в конец файла строку:

```
set(0,'DefaultAxesFontName','Tahoma').
```

## Руководство пользователя по написанию m-файлов

### Создание

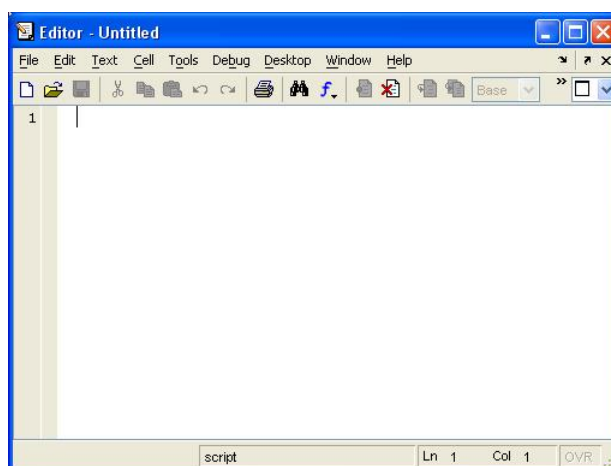


Рис. 2: Окно редактора M-файлов Matlab

M-файлы (программы, необходимые для работы программы) можно создавать двумя способами:

- из рабочей среды Matlab
- из самой программы **BaseOfModelling**

#### Способ 1:

Раскройте меню **File** рабочей среды **Matlab** и в пункте **New** выберите подпункт **M-file** или нажмите кнопку **New M-file** на панели инструментов рабочей среды. Новый файл открывается в редакторе m-файлов, который приведен на рис. 2.

#### Способ 2:

Раскройте меню **Файл** и выберите пункт **Создать**. Откроется тот же редактор М-файлов (см. рис. 2).

## Ввод текста программы

Примером m-файла может служить следующий текст:

```
function xout = pendulum(x)
%
% example of pendulum
%
xout    = x;
xout(1) = x(2);
xout(2) = -sin(x(1));
```

где *pendulum*- имя файла, которое будет вводиться в программе,  $x$ ,  $xout$  - переменная-вектор, строка со знаком % - строка комментария, поясняющая назначение функции. Данный пример моделирует поведение физического маятника, заданного системой:  $x'(t) = Fx(t)$ , где  $x = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$ ,

$$F = \begin{bmatrix} 0 & 1 \\ -\frac{g}{L} & 0 \end{bmatrix} \text{ и } x_0 = \begin{bmatrix} \theta_0 \\ \omega_0 \end{bmatrix}$$

## Сохранение

Пока файл имеет имя Untitled, т.е. ему не присвоено конкретного имени. Для того, чтобы сохранить файл с именем pendulum.m в рабочем каталоге выберите в меню **File** редактора пункт **Save as**.

## Исправление ошибок

Если в m-файле при наборе сделана ошибка, то она выявится в процессе исполнения. Matlab выполняет команды до неправильно введенной, после чего в командное окно выводится сообщение об ошибке. Например, введем следующие строки в файл mydemo.m

```
y=[1 2 3]
```

```
z=y*y
```

```
x=y
```

Очевидно, что во второй строке допущена ошибка. Попытка выполнения такого файла приведет к выводу в командное окно следующего сообщения с гиперссылкой на место ошибки:

```
??? Error using ==> mtimes Inner matrix dimensions must agree.
```

```
Error in ==> mydemo at 2
```

```
z=y*y
```

Щелчок мыши по гиперссылке с именем m-файла делает окно редактора активным и помещает курсор в строку с ошибкой.

## Моделирование в BaseOfModelling

### SyS→ContSyS

Программы, находящиеся в каталоге SyS→ContSyS используются при моделировании систем, непрерывных во времени.

Для начала моделирования необходимо заполнить поля

- имя файла, в котором описана функция
- начальный вектор
- начальный момент
- конечный момент

Все поля заполнены значениями по умолчанию. Т.е. можно сразу «построить» графики движения, фазовое пространство. Для правильной работы программы необходимо, заполнив вышеперечисленные поля, указать размерность задачи.

**Имя файла:** нажатием кнопки  открыть диалоговое окно и выбрать имя m-файла из текущего каталога (см. рис. 3).

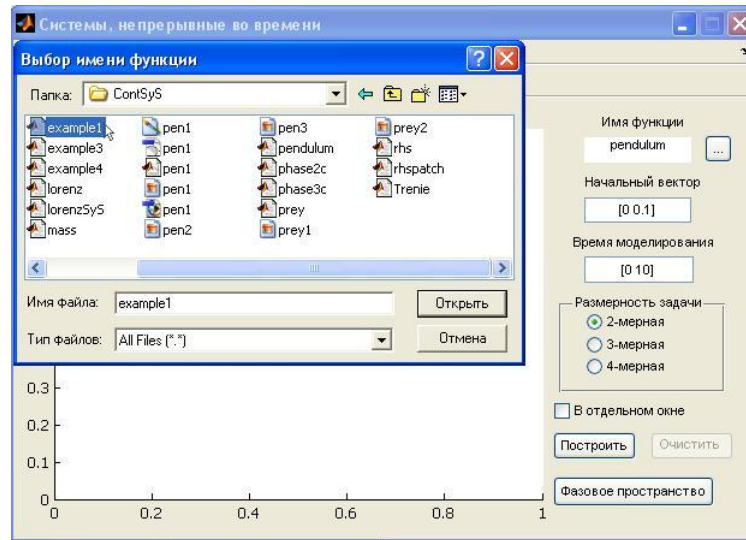


Рис. 3: Выбор имени m-файла

**Начальный вектор:** Это поле заполняется согласно грамматике (правилам записи) в Matlab, т.е. для ввода вектора  $x = (2, 3)$  нужно набрать  $[2\ 3]$  через пробел компоненты, заключив их в  $[ ]$ .

**Начальный момент моделирования:** В большинстве случаев он не нуждается в корректировке.

**Конечный момент:** Момент окончания моделирования.

По окончании моделирования на экран выводятся графики зависимости координат  $x_i$  от  $t$ . Нажатием кнопки «Фазовое пространство» рисуется плоскость  $x_1x_2$  и векторное поле — для 2-мерной задачи и пространство  $x_1x_2x_3$  — для 3-мерной.

Построение графиков в отдельном окне возможно нажатием галочки «В отдельном окне».

Чтобы очистить координатную ось достаточно нажать кнопку «Очистить».

Например, нужно изучить динамику движения математического маятника. Она описывается системой вида  $x' = f(x)$ , где  $f \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -\sin(x_1) \end{bmatrix}$ .

Подготовим m-файл, который вычисляет функцию  $f$ . Такой файл содержится в каталоге ContSys5 и называется pendulum.m:

```
function xout = pendulum(x)
```

```

%
% pendulum example
%
xout      = x;
xout(1)   = x(2);
xout(2)   = -sin(x(1));

```

Чтобы вычислить траекторию системы, заполним:

**Имя файла:** pendulum

**Начальный вектор:** [0.1 0]

**Время моделирования:** [0 30]

выберем пункт 2-мерная задача.

Теперь можно нажать кнопку «Построить» (см. рис. 4).

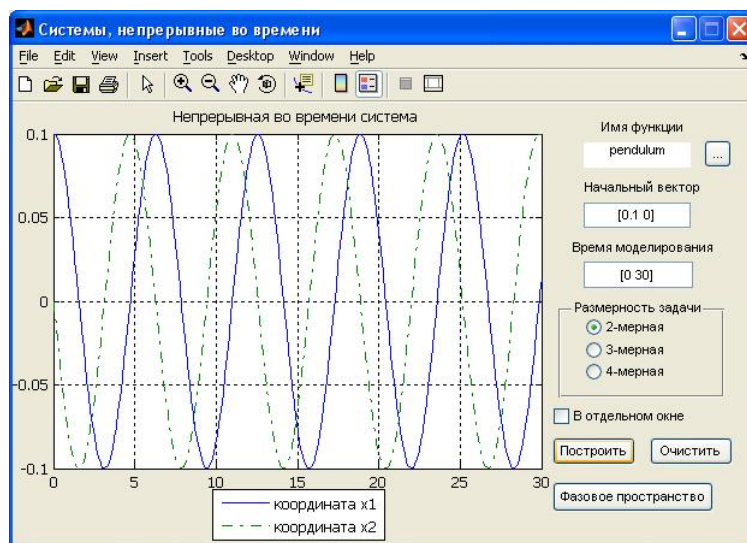


Рис. 4: Моделирование системы, записанной в файле pendulum.m

Для изучения фазовой плоскости (или фазового пространства) нажмем кнопку «Фазовое пространство» (см. рис. 5).

## SyS→DiscrSyS

Программы, находящиеся в каталоге sys→DiscrSyS используются при моделировании систем, дискретных во времени.

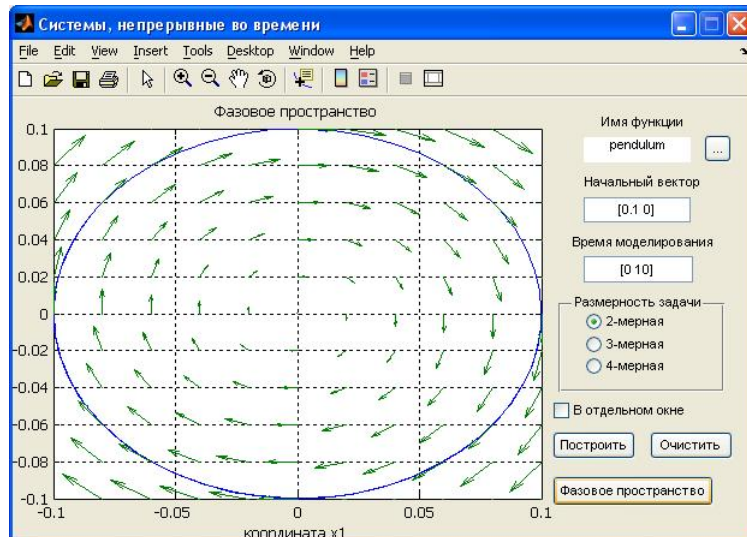


Рис. 5: Фазовое пространство для системы, записанной в файле pendulum.m

Они схожи с подпрограммами каталога ContSyS. Все данные вводятся аналогично. (Конечный момент — число итераций.)

Например, нужно изучить динамику движения системы уравнений вида  $x(k+1) = f(x(k))$ , где  $f \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -0.64 & -0.48 \\ 0.48 & -0.48 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix}$ .

Подготовим m-файл, который вычисляет функцию  $f$ . Такой файл содержится в каталоге DiscrSyS и называется lin2f.m:

```
function y = lin2f(x)
%
A = [
    -0.6400    -0.4800
     0.4800    -0.6400
];
b = [1;1];
y = A*x + b;
```

Чтобы вычислить траекторию системы, заполним:

**Имя файла:** lin2f

**Начальный вектор:** [2 3]

**Время моделирования:** [0 10]



выберем пункт 2-мерная задача.

Теперь можно нажать кнопку «Построить» (см. рис. 6).

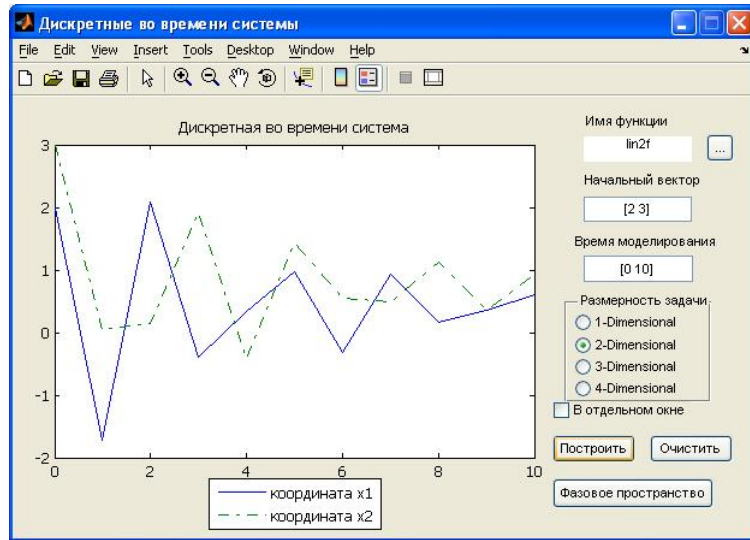


Рис. 6: Моделирование системы, записанной в файле lin2f.m

Для изучения фазовой плоскости (или фазового пространства) нажмем кнопку «Фазовое пространство» (см. рис. 7).

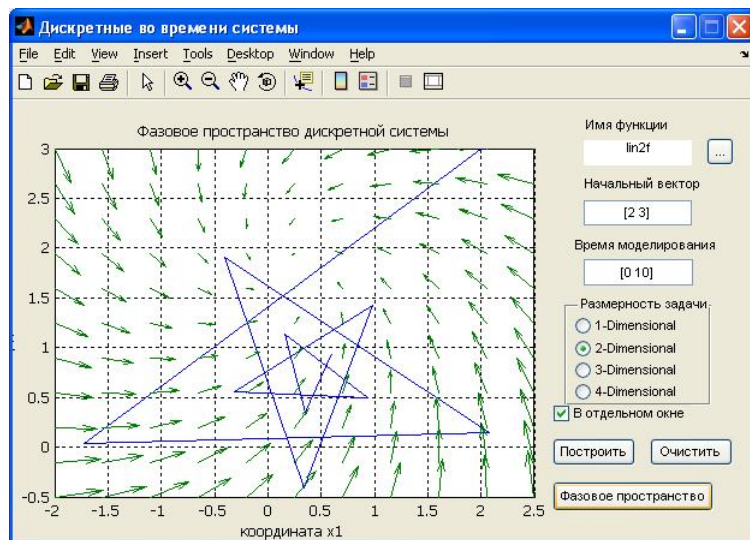


Рис. 7: Фазовое пространство для системы, записанной в файле lin2f.m

## SyS→GradSyS

Программы, находящиеся в каталоге SyS→GradSyS используются при моделировании градиентных систем, непрерывных во времени.

Для начала моделирования необходимо заполнить поля

- имя файла, в котором описана функция
- имя файла, в котором описана функция-градиент
- начальный вектор
- начальный момент
- конечный момент

Все поля заполнены значениями по умолчанию. Редактирование данных аналогично предыдущим пунктам (выбирается два имени файлов — с самой функцией и ее градиентом).

По окончании моделирования на экран выводятся: векторное поле, линии контура, соответствующие кривым, на которых функция  $f$  является константой, траектория системы, проекция функции  $f$  на плоскость  $x_1x_2$ .

Например, нужно изучить динамику движения системы уравнений вида  $x' = -\nabla f(x)$ , где  $f \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \sin(x_1)\sin(x_2) + 1.1$ .

$$\text{Тогда } \nabla f(x) = \begin{bmatrix} -\cos x_1 \sin x_2 \\ -\cos x_2 \sin x_1 \end{bmatrix}$$

Подготовим m-файлы, которые вычисляет функцию  $f$  и  $\nabla f(x)$ . Такие файлы содержатся в каталоге GradSyS и называются fegg.m и gegg.m:

```
function y = fegg(x)
%
y = sin(x(1)) * sin(x(2)) + 1.1 ;
```

```
function y = gegg(x)
%
y = x;
y(1) = cos(x(1))*sin(x(2));
y(2) = sin(x(1))*cos(x(2));
```

Чтобы вычислить траекторию системы, заполним:

**Имя функции:** fegg

**Имя градиента:** gegg

**Начальный вектор:** [1.5 -0.1]

**Время моделирования:** [0 10]

Теперь можно нажать кнопку «Построить» (см. рис. 8).

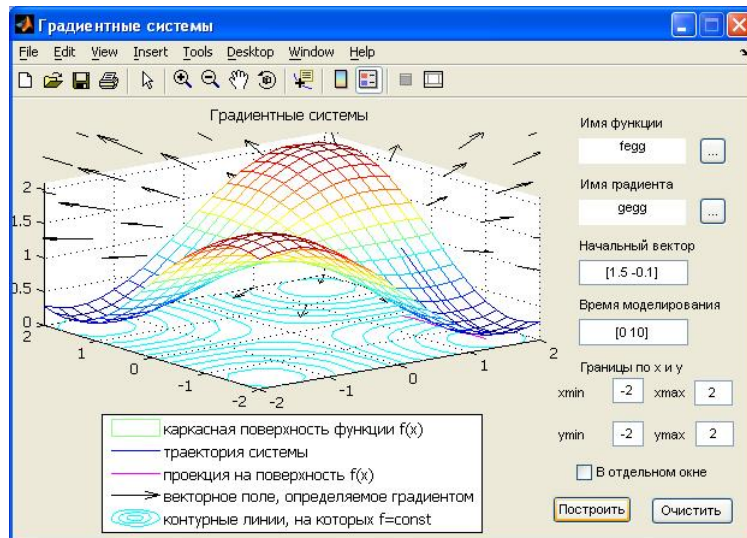


Рис. 8: Моделирование системы, записанной в файле gegg.m

## Bif

Для начала моделирования необходимо заполнить поля

- имя файла, в котором описана функция
- область изменения параметра
- область изменения переменной

Например, нужно изучить динамику движения системы вида  $x' = f_a(x)$ , где  $a$  — параметр,  $f(x) = x^2 + a$ .

Подготовим m-файл, который вычисляет функцию  $f_a$ . Такой файл содержится в каталоге Bif и называется quadrat.m:

```
function y = quadrat(x,a)
```

%

$y = x .* x + a;$

Чтобы вычислить траекторию системы, заполним:

**Имя функции:** `quadrat`

**область изменения параметра:** `[-1.5 0.2]`

**область изменения переменной:** `[-2 2]`

Теперь можно нажать кнопку «Построить» (см. рис. 9).

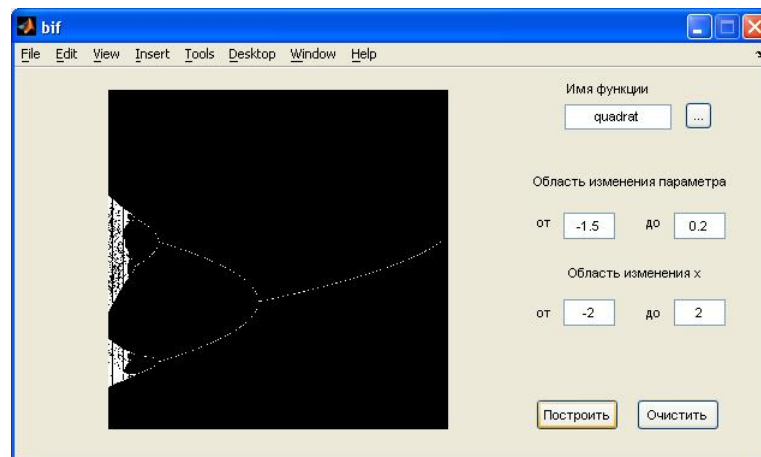


Рис. 9: Моделирование системы, записанной в файле `quadrat.m`

## Fractal→Complex

Программы, находящиеся в каталоге `fractal→complex` используются при построении комплексных динамических систем — алгебраических фракталов.

Перед построением необходимо выбрать тип множества: Множество Мандельброта, Множество Жюлиа или Множество Ньютона и цветовую палитру.

### *Множество Мандельброта*

Для построения множества Мандельброта используется формула  $z_{i+1} = z_i^2 + c$ .

Необходимо заполнить поля

- левая нижняя точка

- правая верхняя точка
- число итераций
- дискретность  $x$

Все поля заполнены значениями по умолчанию. Т.е. можно сразу «построить» изображение. Через пару секунд (1-2 минуты — в зависимости от числа итераций и дискретности  $x$ ) на экране появится изображение. Для него можно поменять цветовую палитру, выбрав из списка соответствующую цветовую схему.

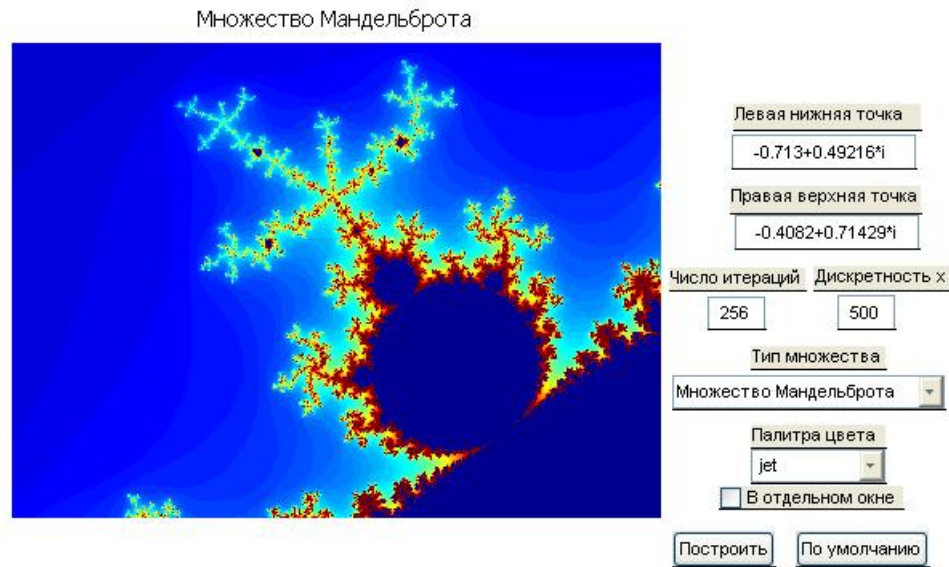


Рис. 10: Множество Мандельброта:  $-0.713 < Re(c) < -0.4082$ ,  $0.49216 < Im(c) < 0.71429$

Например, для построения фрактала Мандельброта (см. рис. 10), нужно заполнить поля

**Левая нижняя точка:**  $-0.713+0.49216i$

**Правая верхняя точка:**  $-0.4082+0.71429i$

**Число итераций:** 256

**Дискретность  $x$ :** 500

**Цветовая палитра:** jet

Теперь можно нажать «построить» (если нужно (ненужно) в отдельном окне, нажать (снять) перед этим галочку в соответствующей строке).

Все изображения можно сохранить на диске, для этого выберите пункт **Save As** меню **File** и введите имя, затем тип рисунка (например, .bmp, .jpg, .eps).

### ***Множество Жюлиа***

Для построения множества Жюлиа используется формула  $z \rightarrow z^2 + c$ , где  $c = const$ .

Все данные для построения множества Жюлиа вводятся аналогично данным множества Мандельброта. Добавляется параметр — начальная точка.

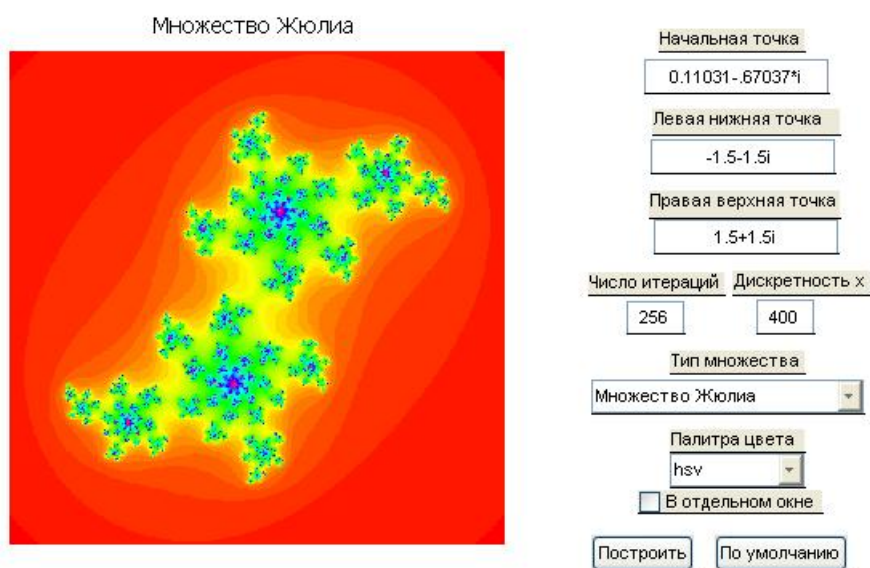


Рис. 11: Множество Жюлиа:  $c = 0.11031 - 0.67037i$

Например, для построения фрактала Жюлиа (см. рис. 11), нужно заполнить поля

**Начальная точка:** 0.11031-.67037i

**Левая нижняя точка:** -1.5-1.5i

**Правая верхняя точка:** 1.5+1.5i

**Число итераций:** 256

**Дискретность :** 400

**Цветовая палитра:** hsv

Теперь можно нажать «построить» (если нужно (ненужно) в отдельном окне, нажать (снять) перед этим галочку в соответствующей строке).

## Множество Ньютона

Для построения множества Ньютона (рис. 12) необходимо ввести число итераций (нечетное число).

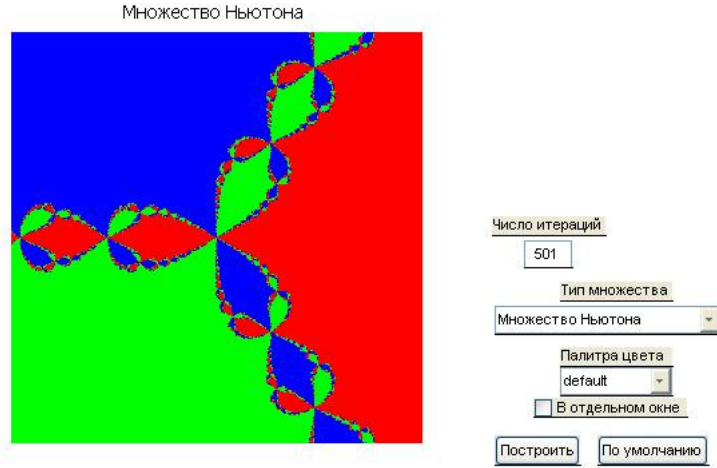


Рис. 12: Множество Ньютона для  $f(z) = z^3 - 1$ ,  $n = 501$

## Fractal→IFS

Программы, находящиеся в каталоге fractal→ IFS используются при построении геометрических фракталов с помощью систем итерируемых функций.

Перед построением необходимо выбрать имя файла, в котором находится матрица и число точек на экране.

Все поля заполнены значениями по умолчанию. Т.е. можно сразу «построить» изображение.

Например, для построения фрактала, заданного системой итерируемых функций:

$$A = \begin{bmatrix} 0.00 & 0.00 & 0.00 & 0.16 & 0.00 & 0.00 \\ 0.85 & 0.04 & -0.04 & 0.85 & 0.00 & 1.60 \\ 0.20 & -0.26 & 0.23 & 0.22 & 0.00 & 1.60 \\ -0.15 & 0.28 & 0.26 & 0.24 & 0.00 & 0.44 \end{bmatrix}$$

нужно:



1. написать текстовый файл:

```
0.00 0.00 0.00 0.16 0.00 0.00
0.85 0.04 -0.04 0.85 0.00 1.60
0.20 -0.26 0.23 0.22 0.00 1.60
-0.15 0.28 0.26 0.24 0.00 0.44
```

2. сохранить его с именем, например, fern.txt в текущем каталоге;

3. в поле имя файла нажать кнопку  и выбрать имя созданного файла из списка;

4. ввести число точек в соответствующее поле;

5. нажать построить (см. рис. 13).

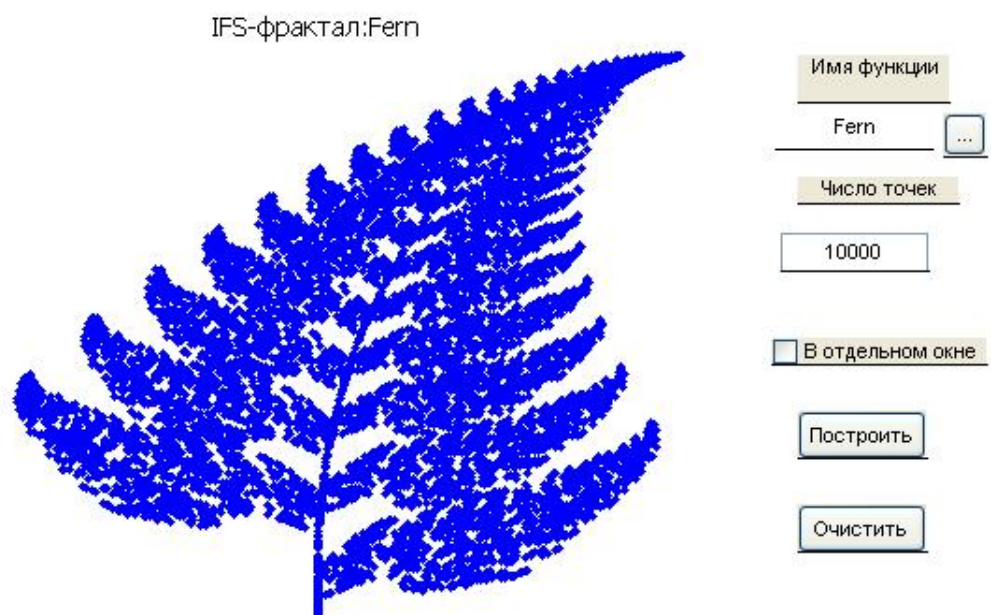


Рис. 13: Фрактал, полученный с помощью системы итерируемых функций из файла fern.txt

В каталоге IFS находится примерный набор файлов для построения фракталов с помощью систем итерируемых функций.



Результаты моделирования в ContSys и DiscrSys (траектории движения) можно сравнить с аналитическим решением, достаточно нажать «Проверить».

Из меню **Справка** программы BaseOfModelling доступно данное руководство пользователя.