

# Исследование структуры множества Мандельброта с использованием программы, написанной с помощью технологии распределенных вычислений NVIDIA CUDA

Олег Пономарев

19 января 2010 г.

## 1 Введение

Множество Мандельброта вопреки распространённому мнению является не просто красивым фракталом, а несёт в себе глубокий смысл, характеризуя динамическую систему, заданную каким-либо отображением комплексной плоскости. Всем известная "красивая фрактальная картинка" является изображением множества Мандельброта для семейства отображений  $Z \rightarrow Z^2 + C$ . Для других отображений оно может выглядеть совсем иначе. Так как множество Мандельброта задаётся математически, мы можем описывать это множество на основе "особых" точек (состояний) динамической системы отображения, не прибегая к построению самого множества. На основе сопоставления численных расчётов динамической системы, полученных в `maple`, и изображения множества Мандельброта в программе `fractal explorer` была выдвинута гипотеза о том, как именно связаны "особые" точки с областями множества. Далее потребовалось проверить эту гипотезу, а также выдвинуть какую-либо более фундаментальную, но для этого требовалось сопоставить достаточно большое количество численных расчётов с изображениями множества. В процессе сопоставления было обнаружено, что программа `Fractal Explorer`, а также все её аналоги, которые были найдены, строят корректное изображение множества лишь для семейства отображений, задаваемых мономами вида  $Z \rightarrow Z^n + C$ , да и для них работают не очень быстро. В связи с этим появилась необходимость написания собственной программы для быстрого построения и удобной навигации по множеству. Для быстрого построения была применена технология распределённых вычислений NVIDIA CUDA. Первая часть данной статьи вводит необходимые определения и описывает методы, используемые для описания и построения множества. Вторая часть посвящена уже самой программе.

## 2 Зачем нужно исследовать множество Мандельброта?

В настоящее время множество Мандельброта применяется в достаточно большом количестве различных областей науки. Например, оно представляет интерес для теории фазовых переходов. И действительно, исследование стабильности состояний различных систем с течением времени является важной задачей современной теоретической физики. Также множество Мандельброта достаточно широко применяется для генерации природных ландшафтов, деревьев, облаков и любых других объектов, используемых для виртуализации мира в 3D графике.

## 3 Определение множества Мандельброта

Пусть  $M_n$  - семейство отображений  $f : X^n \rightarrow X^n$  какого-то пространства в себя:

$$x_i \rightarrow f(x_i)$$

Обозначим за  $J_f$  определитель матрицы Якоби для отображения  $f$ :

$$J_f = \det_{n \times n} \frac{\partial f_i}{\partial x_i}$$

Тогда множество Мандельброта  $\mathcal{B}$  определяется как множество всех таких отображений  $f \in M_n$ , для которых выполняются следующие условия:

$$\begin{aligned} J_f(x) &= 0 \\ \frac{\partial J_f}{\partial x_i} &= 0 \end{aligned}$$

Так как это определение является достаточно абстрактным, оно нуждается в дополнительном пояснении. Так как в дальнейшем будут рассматриваться только множества Мандельброта для отображений вида  $x \rightarrow f(x, C)$ , положим  $M_n = f(x, C)$ , где  $C$  - какой-то произвольный параметр. Так как принадлежность отображения множеству Мандельброта в этом случае определяется лишь значением параметра, то мы можем графически изобразить множество Мандельброта в пространстве параметра  $C$ . Теперь разберёмся с условиями принадлежности множеству Мандельброта. На примитивном уровне их можно интерпретировать так: если мы рассмотрим любое ограниченное множество возможных значений  $x$  и подвергнем их преобразованию  $f(x, C_0)$ , то полученное множество также будет ограниченным. В дальнейшем для простоты изложения будет рассматриваться множество Мандельброта для семейств отображений комплексной плоскости. На самом деле, дальнейшие рассуждения несложно обобщаются и на случай произвольного  $M_n$ .

## 4 Связь структуры множества Мандельброта с орбитами динамической системы

Рассмотрим какое-либо отображение комплексной плоскости  $Z \rightarrow f(Z, C)$ . Назовём точки, переходящие сами в себя после  $n$  таких итераций орбитами  $n$ -ного порядка. Будем использовать обозначение  $F_n$  для многочленов, корни которых задают зависимость точек орбит  $n$ -ного порядка от параметра  $C$ :

$$F_n := f^{on}(Z, C) - Z$$

Понятно, что каждая из орбит порядка  $n$  является также орбитой порядка  $m * n$ , а значит  $F_{m*n}$  описывает также все корни  $F_n$ , а значит  $F_{m*n}/F_n$  - тоже многочлен. Обозначим за  $G_n$  многочлен, корни которого задают орбиты с минимальным порядком  $n$ . Таким образом, мы получаем, что  $G_1 = f_1$ ,  $G_2 = F_2/G_1$ ,  $G_3 = F_3/G_1$ ,  $G_4 = F_4/G_2$ , ...,  $G_n = F_n/\Pi(G_k)$ , где  $k$  являются делителями  $n$ .

### 4.1 Вычисление орбит для отображения $Z \rightarrow Z^2 + C$

Рассмотрим данный конкретный пример для отображения  $Z \rightarrow Z^2 + C$ :

$$F_1 = Z^2 + C - Z = Z^2 - Z + C$$

$$F_2 = (Z^2 + C)^2 + C - Z = Z^4 + 2 * C * Z^2 - Z + C^2 + C$$

$$F_3 = ((Z^2 + C)^2 + C)^2 + C - Z = Z^8 + 4 * C * Z^6 + 6 * C^2 * Z^4 + 2 * C * Z^4 + 4 * C^3 * Z^2 + 4 * C^2 * Z^2 - Z + 2 * C^3 + C^2 + C + C^4$$

$$F_4 = (((Z^2 + C)^2 + C)^2 + C)^2 + C - Z$$

Теперь вычислим  $G_n$ :

$$G_1 = F_1 = Z^2 - Z + C$$

$$G_2 = F_2/G_1 = Z^2 + Z + C + 1$$

$$G_3 = F_3/G_1 = Z^6 + Z^5 + Z^4 + 3 * C * Z^4 + 2 * C * Z^3 + Z^3 + 3 * C^2 * Z^2 + Z^2 + 3 * C * Z^2 + C^2 * Z + Z + 2 * C * Z + C + C^3 + 2 * C^2 + 1$$

$$G_4 = F_4/(G_1 * G_2)$$

### 4.2 Точки обнуления дискриминантов орбит

Посмотрим, при каких значениях  $C$  зануляются дискриминанты  $G_n$ . Понятно, что эти значения  $C$  соответствуют точкам, при которых совпадают пары корней орбит  $n$ -ного порядка совпадают для какого-либо  $C$ . Удивительно совпадение, что именно эти  $C$  соответствуют "крайним" точкам областей множества Мандельброта. Почему это так, станет

понятно ниже. Сначала вычислим дискриминанты  $G_n$  для отображения  $Z \rightarrow Z^2 + C$ :

$$D(G_1) = -4 * C + 1$$

$$D(G_2) = -4 * C - 3$$

$$D(G_3) = -16384 * C^7 - 94208 * C^6 - 208896 * C^5 - 247296 * C^4 - 206976 * C^3 - \\ - 131712 * C^2 - 48020 * C - 16807$$

$$D(G_4) = 1099511627776 * C^{20} + 10995116277760 * C^{19} + 50165218017280 * C^{18} + \\ + 146235046494208 * C^{17} + 319962178650112 * C^{16} + 562159679438848 * C^{15} + \\ + 813773896024064 * C^{14} + 1001745287217152 * C^{13} + 1068184169873408 * C^{12} + \\ + 991615808176128 * C^{11} + 825730309029888 * C^{10} + 617014716530688 * C^9 + \\ + 414422569451520 * C^8 + 263693441433600 * C^7 + 143192862720000 + \\ + C^6 + 74978058240000 * C^5 + 36830829600000 * C^4 + 11179944000000 * C^3 + \\ + 78414482500000 * C^2 + 1037970703125$$

Найдём значения  $C$ , соответствующие нулям дискриминантов. Будем обозначать множества этих значений  $D_n$

$$D_1 = \{0.25\}$$

$$D_2 = \{-0.75\}$$

$$D_3 = \{-0.125 + 0.64954 * I, -0.125 - 0.64954 * I, -1.75\}$$

$$D_4 = \{-1.25, 0.25 + 0.5 * I, 0.25 - 0.5 * I, -1.9406, -0.15472 - 1.0311 * I, \\ - 0.15472 + 1.0311 * I\}$$

$$D_5 = \{-1.9854, -1.8606, -1.6244, -1.2568 - 0.37975 * I, -1.2568 + 0.37975 * I, \\ - 0.48178 - 0.53165 * I, -0.48178 + 0.53165 * I, -0.19787 - 1.1003 * I, \\ - 0.19787 + 1.1003 * I, -0.045062 - 0.98680 * I, -0.045062 + 0.98680 * I, \\ 0.35678 - 0.32858 * I, 0.35678 + 0.32858 * I, \\ 0.35992 - 0.64150 * I, 0.35992 + 0.64150 * I\}$$

Обозначим эти точки на плоскости  $Re(Z), Re(C)$  на рисунке 1.

### 4.3 Точки обнуления результатов орбит

Понятно, что если орбиты разного порядка, например,  $n$  и  $m$  имеют общие точки при каких-либо значениях  $C$ , то эти значения  $C$  соответствуют равенству нулю результата многочленов (в данном случае  $G_n$  и  $G_m$ ), задающих эти орбиты. Посчитаем результаты

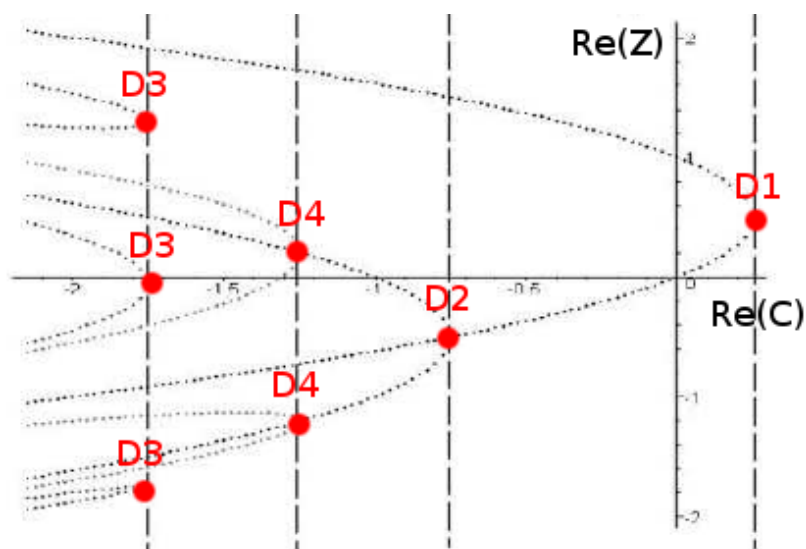


Рис. 1: Изображение орбит и точек равенства нулю дискриминантов

для вычисленных нами многочленов:

$$R(G_1, G_2) = 3 + 4 * C$$

$$R(G_1, G_3) = 16 * C^2 + 4 * C + 7$$

$$R(G_1, G_4) = 16 * C^2 - 8 * C + 5$$

$$R(G_1, G_5) = 256 * C^4 + 64 * C^3 + 16 * C^2 - 36 * C + 31$$

$$R(G_2, G_3) = 1$$

$$R(G_2, G_4) = 16 * C^2 + 40 * C + 25$$

$$R(G_2, G_5) = 1$$

$$R(G_3, G_4) = 1$$

$$R(G_3, G_5) = 1$$

$$R(G_4, G_5) = 1$$

Мы получили, что  $R_{2,3}$  и  $R_{3,4}$  никогда не бывают равными нулю. Из этого следует, что орбиты этих порядков никогда не пересекаются. Найдём же значения  $C$ , при которых

бывают равными нулю остальные результаты. Обозначим множества таких  $C$  как  $R_{n,m}$ :

$$R_{1,2} = \{-0.75\}$$

$$R_{1,3} = \{-0.125 + 0.64954 * I, -0.125 - 0.64954 * I\}$$

$$R_{1,4} = \{0.25 + 0.5 * I, 0.25 - 0.5 * I\}$$

$$R_{1,5} = \{-0.48178 - 0.53165 * I, -0.48178 + 0.53165 * I, 0.35678 - 0.32858 * I, \\ 0.35678 + 0.32858 * I\}$$

$$R_{2,4} = \{-1.25\}$$

Интересным является тот факт, что эти общие точки пересечения орбит соответствуют точкам пересечения областей множества Мандельброта. Почему это так, будет показано ниже. Отметим эти, а также точки равенства нулю дискриминантов на множестве Мандельброта. На основе данных отметок можно пронумеровать области множества согласно тем "особым" точкам этой области, в которых либо только дискриминант, либо и дискриминант, и результат равны нулю. Таким образом, каждая область множества Мандельброта будет задаваться последовательностью чисел, а всё множество будет представлять из себя "лес" областей, состоящий из бесконечного числа "деревьев". Чтобы не перегружать рисунок, отметим на нём лишь последние 2 числа из этой последовательности или 1 число, если эта область является "корнем дерева".

#### 4.4 Нахождение уравнений, задающих границы множества Мандельброта

Рассмотрим точки, являющиеся орбитами  $n$ -ного порядка какого-либо отображения. Рассмотрим траектории этих точек за  $n$  итераций. Очевидно, что полученные точки будут находиться в ограниченной области, а, значит, все эти точки принадлежат множеству Мандельброта согласно конкретизированному для нашего случая определению. Аналогичные рассуждения можно применить к точкам, лежащим "близ" точек орбит, если эти точки после многократного применения отображения продолжают оставаться "близ" точек орбит. Иными словами, рассмотрим устойчивость точек орбит. Возьмём какое-либо уравнение, задающее точки орбит порядка  $n$  -  $G_n(Z, C)$ . Пусть  $Z_n(C)$  - функция, определяющая точки орбит порядка  $n$  в зависимости от значения параметра  $C$ :

$$Z_n(C) : G_n(Z_n(C), C) = 0$$

Тогда внесём в уравнение  $F_n(Z_n(C), C) + Z$ , определяющее положение точки после  $n$  итераций, малое отклонение  $\delta Z$  и преобразуем выражение к более удобному виду, разложив его в ряд около точки орбиты до первого порядка малости:

$$F_n(Z_n(C) + \delta Z, C) + Z = F_n(Z_n(C), C) + F'_n(Z_n(C), C)\delta Z + 1 = F'_n(Z_n(C), C)\delta Z + 1$$

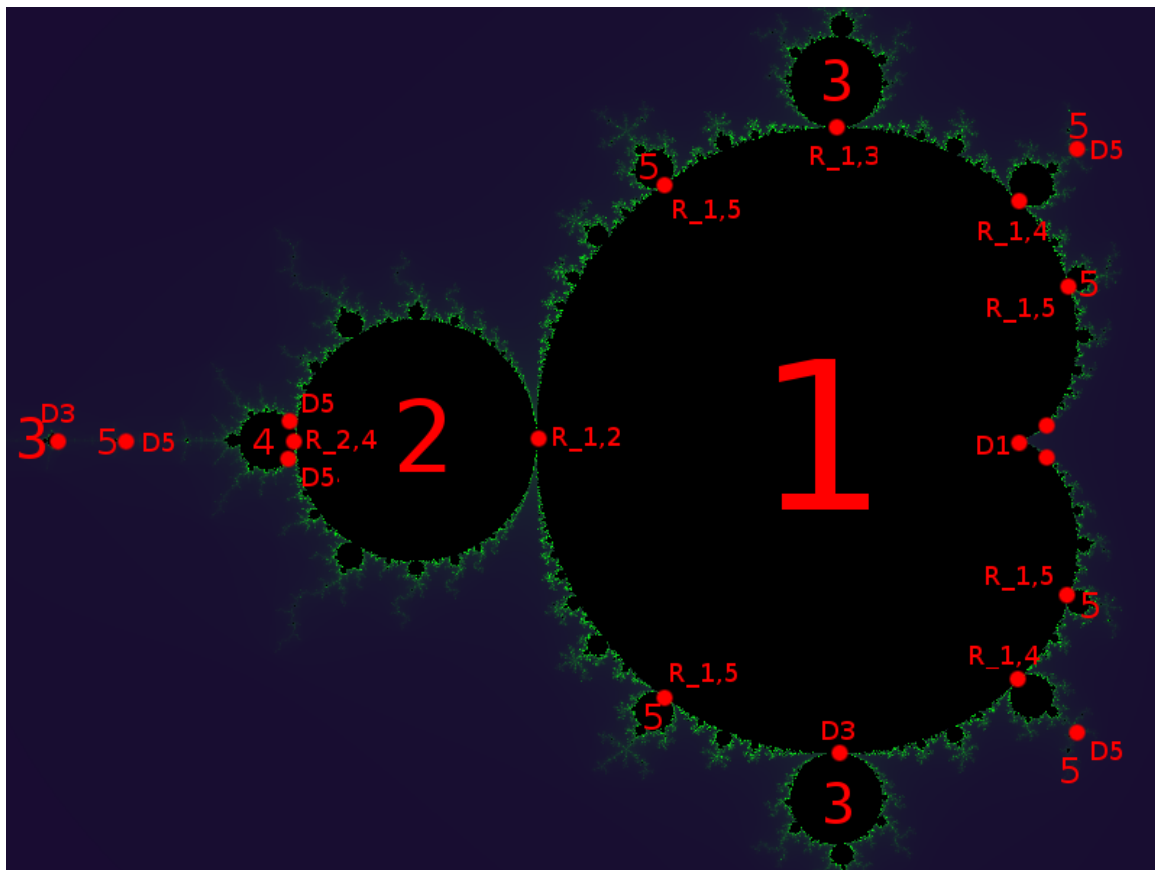


Рис. 2: Изображение орбит и точек равенства нулю дискриминантов

Так как такого рода итерации могут выполняться бесконечное число раз, понятно, что то, "улетит" ли точка от орбиты, зависит только от модуля производной  $F'_n(Z_n(C), C) + Z$ , а именно от того, будет ли он больше или меньше 1. Если же  $|F'_n(Z_n(C), C) + 1| > 1$ , то в пределе точка  $Z_n(C) + \delta Z$  окажется вне ограниченного участка, т. е. не будет принадлежать множеству Мандельброта. Аналогично получаем то, что точки, в которых  $|F'_n(Z_n(C), C) + 1| = 1$  задают границу области, получаемой из орбит порядка  $n$ , множества Мандельброта. Если модуль комплексного числа равен 1, то оно представимо в виде  $e^{i\phi}$ , т. е. уравнение границы области, полученной из орбиты порядка  $n$ , множества Мандельброта задаётся уравнением  $F'_n(Z_n(C), C) + 1 = e^{i\phi}$  или, если избавиться от лишних обозначений:

$$\begin{cases} G_n(Z, C) = 0 \\ F'_n(Z, C) + 1 = e^{i\phi} \end{cases}$$

## 5 Метод вычисления точек множества Мандельброта

Рассмотрим какое-либо значение параметра  $C_0$ , задающее какое-либо конкретное отображение  $Z \rightarrow f(Z, C_0)$ . Утверждается, что для того, чтобы узнать, принадлежит ли это отображение множеству Мандельброта, нужно проделать следующие действия:

- Найти все  $Z_0$ , являющиеся решениями уравнения  $f'(Z, C_0) = 0$
- Найти пределы последовательностей  $f^{on}(Z_0, C_0)$  для каждого значения  $Z_0$
- Если хотя бы один из этих пределов не равен  $\infty$ , то отображение, задаваемое данным  $C_0$ , принадлежит множеству Мандельброта

Разберёмся в том, почему же этот способ работает. В первом пункте мы выбираем из всех точек  $Z$  аттракторы (центры притяжения), т. е. такие точки, в которые переходят все близлежащие точки после проведения первого отображения ( $Z_0 + \delta Z \rightarrow f(Z) + \delta Z f'(Z_0) + \dots$ ). Понятно, что при многократном применении отображения все точки исходного пространства можно будет разделить на зоны влияния разных аттракторов. Все точки, которые не относятся к аттрактору  $Z_0 = \infty$ , будут оставаться в какой-либо ограниченной области, а, значит, согласно конкретизированному определению множеств Мандельброта принадлежат этому множеству. Именно то, относятся ли точки к аттрактору  $Z_0 = \infty$ , и проверяет второе условие. Таким образом, наш метод для определения принадлежности отображения, задаваемого параметром  $C$  множеству Мандельброта, является корректным.

## 6 Постановка задачи

Уже можно было понять, что многие факты о множестве Мандельброта выдвигались на уровне интуитивно понятных гипотез, а лишь потом находились способы строгого доказательства этих гипотез. В связи с этим на начальной стадии выдвижения каждой гипотезы требовалась проверка её корректности. Для такой проверки была необходима программа, обладающая следующими возможностями:

- Построение множества Мандельброта для любого семейства отображений комплексной плоскости вида  $Z \rightarrow f(Z) + C$
- Удобная навигация по построенному множеству
- Возможность введения дополнительного комплексного параметра  $T$  в формулу, задающую семейство отображений, который можно будет быстро и удобно изменять
- Возможность динамического регулирования точности построения

Также было решено, что программа должна быть:



- Кроссплатформенной
- Легкорасширяемой
- Быстрой настолько, насколько это возможно на данной машине

Всё это необходимо для того, чтобы обеспечить удобное использование программы. Кроссплатформенность необходима, так как в настоящее время всё больше людей используют ОС GNU/LINUX, но в то же время всё ещё достаточно много пользователей ОС MICROSOFT WINDOWS. Лёгкая расширяемость необходима из-за того, что исследования множества Мандельброта продолжаются и ещё не ясно, как изменятся требования к программе в будущем. Быстрой программа должна быть потому, что пользователю может потребоваться высокая точность расчётов множества, но при этом время выполнения должно оставаться приемлемым. Иными словами, программа должна полностью использовать доступные ей ресурсы компьютера.

## 7 Архитектура программы

### 7.1 Основные компоненты программы

Для реализации данной задачи был написан программный комплекс, состоящий из следующих основных компонентов:

- Математический модуль, просчитывающий точки множества Мандельброта
- Графический модуль, занимающийся отображением и навигацией по множеству Мандельброта
- Модуль графического интерфейса пользователя, позволяющий динамически изменять параметры вычисления и отображения множества Мандельброта

Рассмотрим каждый модуль подробнее:

### 7.2 Математический модуль

Конечной задачей математического модуля является определение точек, задающих отображения, принадлежащие множеству Мандельброта. Входные данные этого модуля - границы области множества Мандельброта, которую надо построить, точность построения, а также семейство отображений, для которых нужно построить множество. Этот модуль можно условно разделить на 3 основных части:

- Распознаватель формулы, задающей семейство отображений
- Модуль, определяющий "особые", точки  $Z$  для итерирования

- Модуль, определяющий принадлежность точек  $S$  множеству Мандельброта

Рассмотрим каждую из этих частей модуля подробнее:

### 7.2.1 Распознаватель формулы, задающей семейство отображений

Так как семейство отображений вводится пользователем в виде текстовой строки, этот модуль должен распознавать эту строку и генерировать код, который можно было бы вызывать из кода основной программы для итерирования точек. Этот модуль последовательно выполняет следующие действия:

- Проверяет введённую пользователем формулу на наличие ошибок
- Если ошибок не обнаружено, модуль генерирует код, предназначенный для итерирования точек
- Вызывает компилятор для создания библиотеки динамической линковки, которая впоследствии будет подключена к основной программе

Язык сгенерированного кода зависит от конкретной реализации модуля. Среди написанных версий существуют версии, генерирующий код на C++ и на его расширении CUDA C++. Тогда для компиляции кода используются g++ и nvcc соответственно. Такой способ распознавания формулы был выбран из-за простоты реализации. Временем, которое тратится на эти действия, можно пренебречь по сравнению со временем непосредственного вычисления точек множества Мандельброта. Также неоспоримым достоинством данного способа распознавания является то, что на выходе получается машинный код, а это необходимо, так как этот код вызывается многократно в процессе подсчёта точек множества Мандельброта и, по сути, скорость подсчёта множества зависит именно от быстродействия данного кода.

### 7.2.2 Модуль, определяющий "особые" точки $Z$ для итерирования

Целью этого модуля, по сути, является дифференцирование отображения и решение полученного уравнения, т. е. произведение шага 1 из выше приведённого метода расчёта точек множества Мандельброта. Так как данная программа самостоятельно не разбирает формулу, а лишь ищет в ней ошибки, эти 2 действия могли бы являться достаточно трудоёмкими при попытках их реализации на C++. С другой стороны, эти действия выполняются лишь один раз для каждого конкретного семейства отображений, поэтому их быстродействие не очень-то и важно. Именно из-за этих причин было найдено достаточно изящное решение: вызывать для этих целей систему символьных расчётов maple. Использование достаточно "тяжёлого" программного продукта maple не является недостатком данной программы, так как предполагается, что её будут использовать люди, так или иначе имеющие отношение к теоретической физике или какой-либо области математики, а, как правило,

такие люди используют `maple` в своей работе, а, значит, он должен быть уже установлен на их компьютерах и/или в компьютерных классах (именно так и сделано в Институте Теоретической и Экспериментальной Физики).

### 7.2.3 Модуль, определяющий принадлежность точек $S$ множеству Мандельброта

Целью этого метода является исполнение оставшихся двух шагов вышеприведённого метода определения точек множества Мандельброта. Таким образом, он должен многократно итерировать (т.е. действовать на них отображением комплексной плоскости) "особые" точки фазового пространства. Именно из-за необходимости многократного применения отображения к точкам комплексной плоскости (в зависимости от точности, не менее 100 раз для каждой точки отображаемой части плоскости), изначально этот процесс производился в четырёх независимых потоках, чтобы полностью использовать все ресурсы четырёхъядерного процессора. Затем он был переписан для исполнения на видеокарте с использованием технологии NVIDIA CUDA. Этот шаг дал достаточно большой прирост в производительности. Модуль был доработан для возможности параллельного расчёта в 120 независимых потоках. Это необходимо, так как современные видеокарты NVIDIA имеют до 120 процессоров. Таким образом, в соответствии с поставленными задачами, достигается максимально возможное использование ресурсов компьютера.

## 7.3 Графический модуль

Задача этого модуля - изображение и удобная навигация по точкам множества Мандельброта. Также при написании этого модуля необходимо было сохранить кроссплатформенность, поэтому средством его реализации был избран набор кроссплатформенных библиотек Qt. Qt также обеспечило простой способ взаимодействия между графическим модулем, модулем графического интерфейса и математическим модулем, так как реализованная в этих библиотеках сигнально-слотовая модель взаимодействий не требует дополнительных трудозатрат для написания кода синхронизации модулей. В данном модуле имеется возможность передвижения по изображению множества и зуммирования как всего множества, так и какого-либо конкретного его участка. Из-за того, что при написании этого модуля, как и во всей программе, учитывались стратегии лёгкой расширяемости, при желании он может быть использован для отображения любого множества точек, например, множеств решения неравенств. Этот модуль также используется для отображения точек множества Julia. Чтобы получить быстрый способ отрисовки множеств, была использована библиотека OpenGL. Использование OpenGL также позволяет быстро и легко производить трансформации множества.

## 7.4 Модуль графического интерфейса

Этот модуль используется в качестве основного модуля взаимодействия с пользователем. Он помещает графический модуль в окно, а также предоставляет меню и набор "горячих" клавиш для конфигурации таких параметров построения множества Мандельброта, как семейство преобразований, на множестве которых необходимо построить множество Мандельброта, точность построения, задачи значения дополнительного параметра  $T$ , который может употребляться при указании семейства отображений. Этот модуль, как и графический, написан с использованием библиотек Qt.

## 8 Демонстрация программы

Обоснуем неправильность работы программы `fractal explorer`. Для этого построим множество Мандельброта для семейства отображений  $Z \rightarrow Z^4 + Z^3 + Z^2 + C$ . На рисунке 3 показан результат построения, полученный с помощью программы `Fractal Explorer`, на рисунке 4 результат, полученный с помощью авторской программы `TBG-Fractal`. Видно, что рисунки не совпадают. Вычислим точки обнуления дискриминанта уравнения, задающего зависимость неподвижных точек (орбит первого порядка) от параметра отображения  $C$ . Для этого воспользуемся системой символьных вычислений `maple`.

$$\begin{aligned} &> \text{evalf}[4](\text{solve}(\text{discrim}(Z^4 + Z^3 + Z^2 - Z + C, Z) = 0, C)); \\ &0.1749, -0.4409 + 0.8875I, -0.4409 - 0.8875I \end{aligned}$$

В начале статьи показано, что эти точки должны соответствовать особым точкам областей множества Мандельброта (или по крайней мере принадлежать ему). Отметим эти особые точки красным цветом на обоих рисунках. Теперь из рисунков наглядно видно, что `Fractal Explorer` строит некорректное множество. Рассмотрим ещё один интересный пример, который невозможно получить с помощью программы `Fractal Explorer`. Построим множество Мандельброта для семейства отображений  $Z \rightarrow t * Z^3 + (1 - t) * Z^2 + C$ . Интерес для автора представляло изменение множества Мандельброта для этого семейства отображений при различных значениях  $t$ . На рисунке 5 показано, как это множество выглядит при  $t = 0.14 + 0.009I$ .

## 9 Результаты работы

- Был выдвинут и проверен ряд гипотез о структуре множества Мандельброта, с помощью которых можно теоретически описывать области этого множества
- Была выявлена ошибка в построении множества Мандельброта у всех тех программ, которые удалось найти в свободном доступе в интернете (около 5 разных программ)

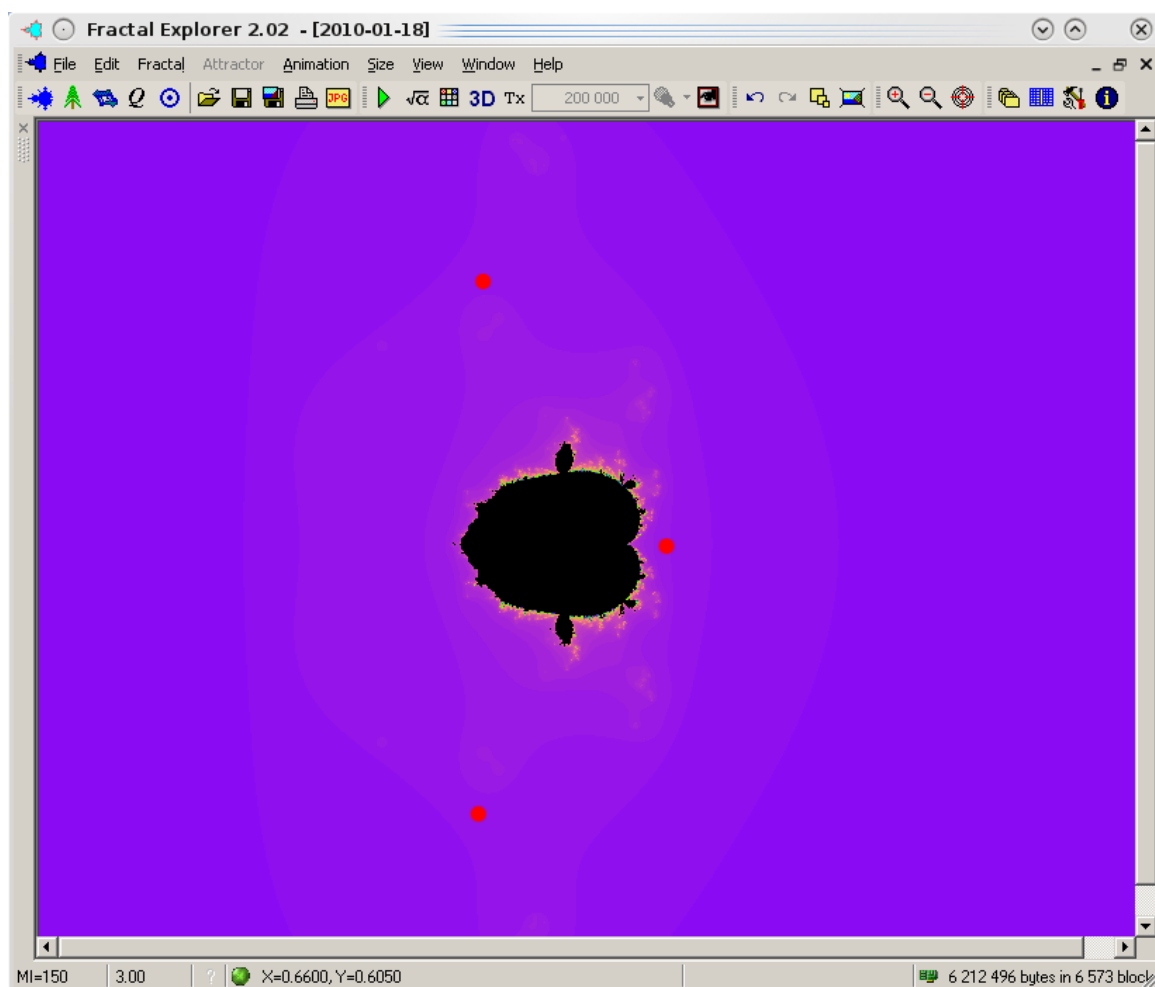


Рис. 3: Некорректное множество Мандельброта для семейства отображений  $Z \rightarrow Z^4 + Z^3 + Z^2 + C$ , построенное с помощью Fractal Explorer

- Для выдвижения и проверки новых гипотез для описания множества Мандельброта была написана собственная программа, полностью отвечающая указанным выше требованиям. Одним из колоссальных достоинств программы является высокая скорость работы, достигнутая за счёт использования видеокарты для математических расчётов. Ещё одно её несомненное достоинство - кроссплатформенность.

## 10 Планы на будущее

- Дальнейшее исследование множества Мандельброта. Выдвижение и проверка новых гипотез

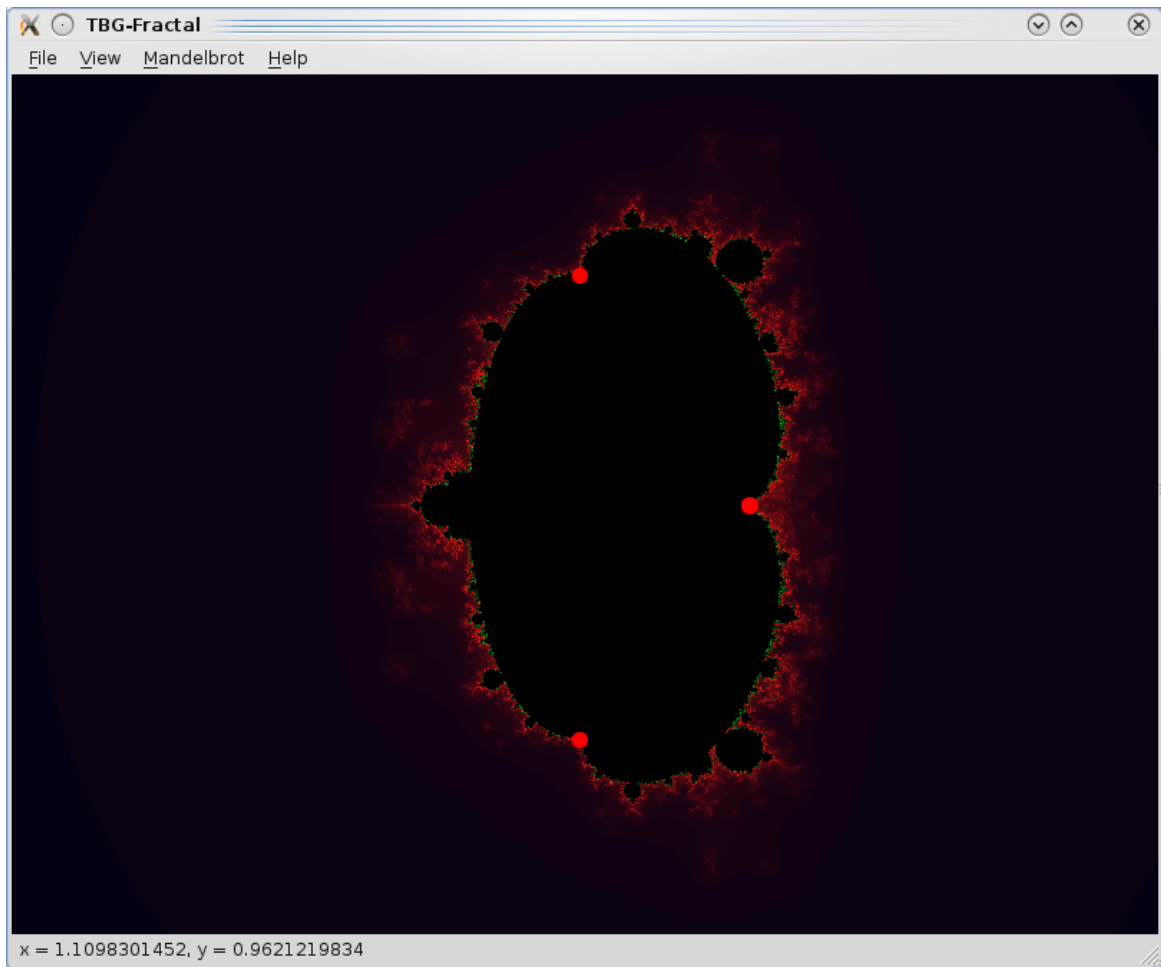


Рис. 4: Корректное множество Мандельброта для семейства отображений  $Z \rightarrow Z^4 + Z^3 + Z^2 + C$ , построенное с помощью авторской программы TBG-Fractal

- Добавление в математический модуль возможности производить расчёты с помощью технологии AMD STREAM для использования вычислительных мощностей видеокарт АТІ.
- Генерация объектов для трёхмерной визуализации мира на основе множеств Мандельброта для различных семейств отображений

Автор статьи выражает благодарность Морозову А.Ю., Морозову А.А., Дединскому И.Р., Власову А.Д. за постановку задачи, помощь в реализации проекта и за проявленный к работе интерес.

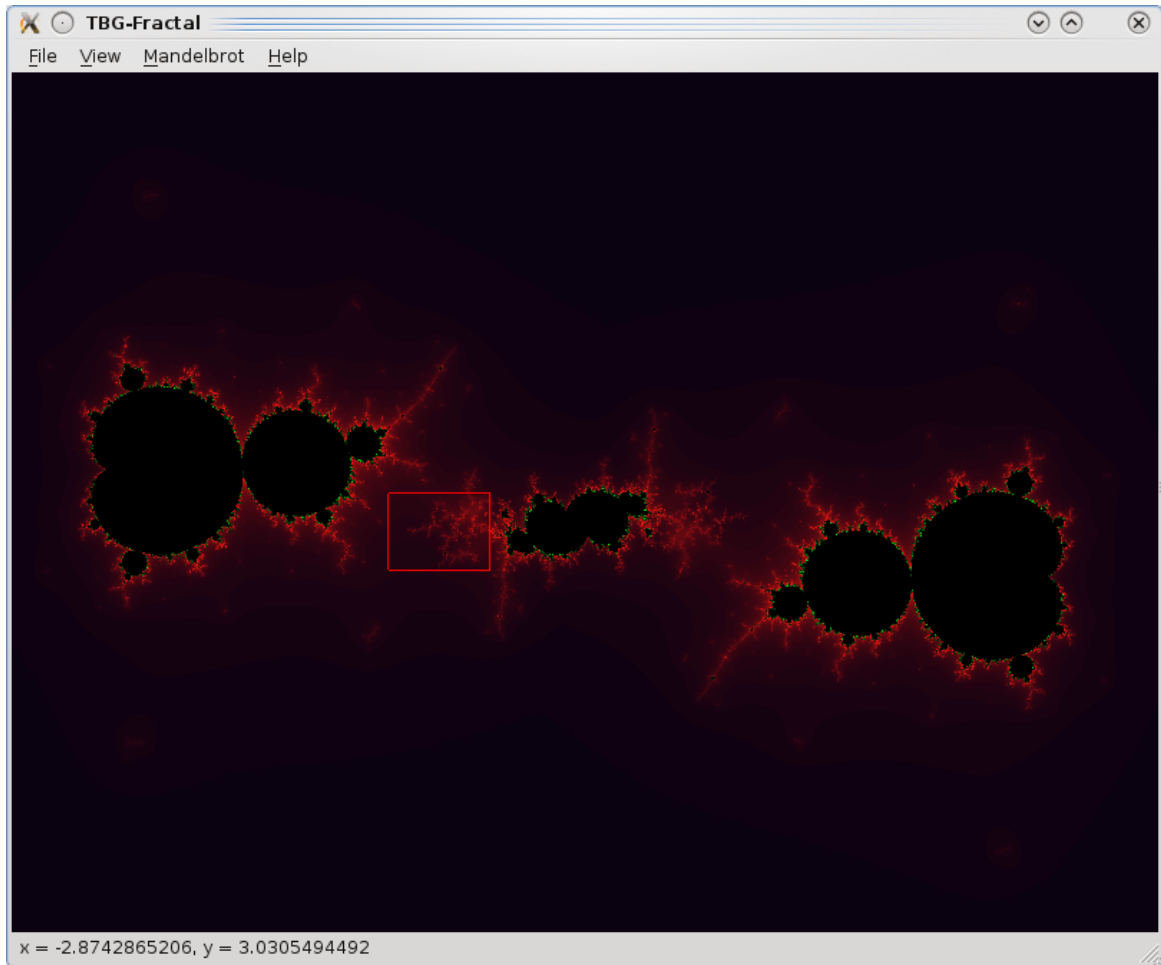


Рис. 5: Множество Мандельброта для семейства отображений  $Z \rightarrow t * Z^3 + (1 - t) * Z^2 + C$  при  $t = 0.14 + 0.009I$ , построенное с помощью авторской программы TBG-Fractal

## Список литературы

- [1] V. Dolotin and A. Morozov, *Introduction to Non-Linear Algebra*, arXiv:hep-th/0609022
- [2] Andrey Morozov *Universal Mandelbrot Set as a Model of Phase Transition Theory*, JETPLett.86:745-748,2007
- [3] V. Dolotin and A. Morozov, *Algebraic Geometry of Discrete Dynamics. The case of one variable*, arXiv:hep-th/0501235v2
- [4] V. Dolotin and A. Morozov, *On the shapes of elementary domains or why Mandelbrot Set is made from almost ideal circles?*, Int.J.Mod.Phys.A23:3613-3684,2008

- [5] I.Gelfand, M.Kapranov and A.Zelevinsky, *Discriminants, Resultants and Multidimensional Determinants* (1994) Birkhauser
- [6] *NVIDIA CUDA Compute Unified Device Architecture Programming Guide*, [www.nvidia.org](http://www.nvidia.org)
- [7] Ричард С. Райт-мл, Бенджамин Липчак *OpenGL. Суперкнига, 3-е издание* (2006) "Вильямс"
- [8] Андрей Алесксандреску *Современное проектирование на C++* (2002) "Вильямс"
- [9] Шлее М. *Qt4. Профессиональное программирование на C++* (2007) БХВ-Петербург