**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ  
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ  
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук  
Департамент программной инженерии

|  |  |
| --- | --- |
| СОГЛАСОВАНО  Старший преподаватель департамента программной инженерии факультета компьютерных наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. А. Шершаков «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. | УТВЕРЖДАЮ  Академический руководитель образовательной программы «Программная инженерия» профессор департамента программной инженерии, канд. техн. наук  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ В. В. Шилов «\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г. |

|  |  |
| --- | --- |
| ***Подп. и дата*** |  |
| ***Инв. № дубл.*** |  |
| ***Взам. инв. №*** |  |
| ***Подп. и дата*** |  |
| ***Инв. № подл*** | RU.17701729.04.01-01 51 01-1-ЛУ |

**ПРОГРАММА-РАСШИРЕНИЕ MICROSOFT VISIO ДЛЯ ИМПОРТА ГРАФОВ В ФОРМАТЕ DOT**

**Программа и методика испытаний**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.01-01 51 01-1-ЛУ**

Исполнитель  
студент группы БПИ173  
\_\_\_\_\_\_\_\_\_\_ / Переплетчиков А. И. /  
«\_\_» \_\_\_\_\_\_\_\_\_\_\_ 2019 г.

**Москва 2019**

УТВЕРЖДЕНRU.17701729.04.01-01 51 01-1-ЛУ

|  |  |
| --- | --- |
| Подп. и дата |  |
| Инв. № дубл. |  |
| Взам. инв. № |  |
| Подп. и дата |  |
| Инв. № подл | RU.17701729.04.01-01 51 01-1-ЛУ |

**ПРОГРАММА-РАСШИРЕНИЕ MICROSOFT VISIO ДЛЯ ИМПОРТА ГРАФОВ В ФОРМАТЕ DOT**

**Программа и методика испытаний**

**RU.17701729.04.01-01 51 01-1**

**Листов 25**

**Москва 2019**

**Содержание**

[**1. Объект испытаний** 4](#_Toc451347246)

[**1.1. Наименование программы** 4](#_Toc451347247)

[**1.2. Область применения** 4](#_Toc451347248)

[**1.3. Обозначение испытуемой программы** 4](#_Toc451347249)

[**2. Цель испытаний** 5](#_Toc451347250)

[**3. Требования к программе** 6](#_Toc451347251)

[**3.1. Требования к функциональным характеристикам** 6](#_Toc451347252)

[**3.1.1. Требования к составу выполняемых функций** 6](#_Toc451347253)

[**3.1.2. Требования к организации входных данных** 6](#_Toc451347254)

[**3.1.3. Требования к организации выходных данных** 6](#_Toc451347255)

[**3.2. Требования к надежности** 6](#_Toc451347256)

[**3.3. Требования к интерфейсу** 6](#_Toc451347257)

[**4. Требования к программной документации** 7](#_Toc451347258)

[**5. Средства и порядок испытаний** 8](#_Toc451347259)

[**5.1. Технические средства, используемые во время испытаний** 8](#_Toc451347260)

[**5.2. Программные средства, используемые во время испытаний** 8](#_Toc451347261)

[**5.3. Порядок проведения испытаний** 8](#_Toc451347262)

[**5.4. Условия проведения испытаний** 8](#_Toc451347263)

[**5.4.1. Климатические условия** 8](#_Toc451347264)

[**5.4.2. Требования к численности и квалификации персонала** 8](#_Toc451347265)

[**6. Методы испытаний** 9](#_Toc451347266)

[**6.1. Испытание выполнения требований к программной документации** 9](#_Toc451347267)

[**6.2. Испытание выполнения требований к интерфейсу** 9](#_Toc451347268)

[**6.3. Испытание выполнения требований к функциональным характеристикам** 11](#_Toc451347269)

[**6.3.1. Испытание выполнения требований к функциональным характеристикам в части задания дифференциального уравнения для решения, границ области, на которой дифференциальное уравнение решается, и граничных условий** 11](#_Toc451347270)

[**6.3.2. Испытание выполнения требований к функциональным характеристикам в части настройки параметров нейронной сети и ее обучения** 11](#_Toc451347271)

[**6.3.3. Испытание выполнения требований к функциональным характеристикам в части решения заданного дифференциального уравнения нейросетевым способом** 11](#_Toc451347272)

[**6.3.4. Испытание выполнения требований к функциональным характеристикам в части вывода визуализированных результатов решения дифференциального уравнения в форме значений искомой функции на заданной области** 20](#_Toc451347273)

[**6.3.5. Испытание выполнения требований к функциональным характеристикам в части прерывания процесса решения дифференциального уравнения во время его выполнения** 21](#_Toc451347274)

[**6.3.6. Испытание выполнения требований к функциональным характеристикам в части решения нового дифференциального уравнения без перезапуска программы** 23](#_Toc451347275)

[**6.4. Испытание выполнения требований к надежности** 25](#_Toc451347276)

# **1. Объект испытаний**

## **1.1. Наименование программы**

Наименование программы: «Программа нейросетевого решения дифференциальных уравнений в частных производных».

## **1.2. Область применения**

Программа будет применяться для решения дифференциальных уравнений, представляющих собой в левой части полиномы из искомой функции, ее производных, коэффициентов, свободной функции, с заданными граничными условиями, с помощью аппроксимации искомой функции, используя нейронную сеть.

## **1.3. Обозначение испытуемой программы**

Наименование темы разработки – DiffEqNeuroSolver.

# **2. Цель испытаний**

Цель испытаний – проверка соответствия функционала и характеристик программного продукта требованиям к программному продукту, изложенным в документе «Техническое задание» (ГОСТ 19.201-78).

# **3. Требования к программе**

## **3.1. Требования к функциональным характеристикам**

### **3.1.1. Требования к составу выполняемых функций**

Программа должна обеспечивать возможность выполнения следующих функций:

1. задание дифференциального уравнения для решения, границ области, на которой дифференциальное уравнение решается, и граничных условий;
2. настройка параметров нейронной сети и ее обучения: количество нейронов в слоях нейронной сети, количество итераций обучения, шаг градиентного спуска;
3. решение заданного дифференциального уравнения нейросетевым способом (с помощью аппроксимации искомой функции, используя нейронную сеть);
4. вывод визуализированных результатов решения дифференциального уравнения в форме значений искомой функции на заданной области;
5. прерывание процесса решения дифференциального уравнения во время его выполнения;
6. решение нового дифференциального уравнения без перезапуска программы.

### **3.1.2. Требования к организации входных данных**

Программа должна позволять вводить входные данные (данные о дифференциальном уравнении для решения, границах области, на которой дифференциальное уравнение решается, и граничных условий, параметрах нейронной сети и ее обучения) через текстовые поля или выпадающие списки окна Windows Forms.

### **3.1.3. Требования к организации выходных данных**

Программа должна выводить результат решения дифференциального уравнения в виде цветового графика искомой функции на заданной области (с возможностью посмотреть значение искомой функции в каждой из отображаемых точек заданной области) и информации о максимальной ошибке нейронной сети после ее обучения.

## **3.2. Требования к надежности**

Программа обеспечивает проверку корректности входных данных.

Для корректной работы программы требуется стабильное и корректное функционирование компьютера и операционной системы.

## **3.3. Требования к интерфейсу**

Программа должна иметь оконный интерфейс Windows Forms с возможностью ввода входных данных и вывода результата в окнах программы.

# **4. Требования к программной документации**

Состав программной документации должен включать в себя следующие компоненты:

1. Техническое задание (ГОСТ 19.201-78)
2. Программа и методика испытаний (ГОСТ 19.301-78)
3. Пояснительная записка (ГОСТ 19.404-79)
4. Руководство оператора (ГОСТ 19.505-79)
5. Текст программы (ГОСТ 19.401-78)

# **5. Средства и порядок испытаний**

## **5.1. Технические средства, используемые во время испытаний**

Для нормального функционирования программы требуется компьютер, оснащенный следующими техническими компонентами:

1. процессор не ниже Intel Pentium/Celeron, AMD K6/Athlon/Duron или совместимый с ними с тактовой частотой не ниже 1 ГГц;
2. 512 Мб ОЗУ или более;
3. жесткий диск с объемом свободной памяти не менее 100 Мб;
4. VGA-совместимые видеоадаптер и монитор с разрешением не ниже 1280х800;
5. клавиатура и мышь.

## **5.2. Программные средства, используемые во время испытаний**

Для нормального функционирования программы требуется компьютер, оснащенный следующими программными компонентами:

1. операционная система Microsoft Windows XP (SP2, SP3) / Vista / 7 / 8 / 8.1 / 10;
2. библиотека Microsoft .NET Framework 4.5 и выше.

## **5.3. Порядок проведения испытаний**

Испытания должны проводиться в следующем порядке:

1. проверка требований к программной документации;
2. проверка требований к интерфейсу;
3. проверка требований к функциональным характеристикам;
4. проверка требований к надежности.

## **5.4. Условия проведения испытаний**

### **5.4.1. Климатические условия**

Климатические условия проведения испытаний программного продукта должны удовлетворять стандартным требованиям к климатическим условиям использования компьютера и использования и хранения соответствующих электронных и бумажных носителей информации.

### **5.4.2. Требования к численности и квалификации персонала**

Для испытаний программы требуется по крайней мере один пользователь.

Требуемая квалификация пользователя программы – оператор ЭВМ с базовыми знаниями в области дифференциальных уравнений и нейронных сетей.

# **6. Методы испытаний**

Испытания проводятся в порядке, указанном в п. 5.3 настоящего документа.

## **6.1. Испытание выполнения требований к программной документации**

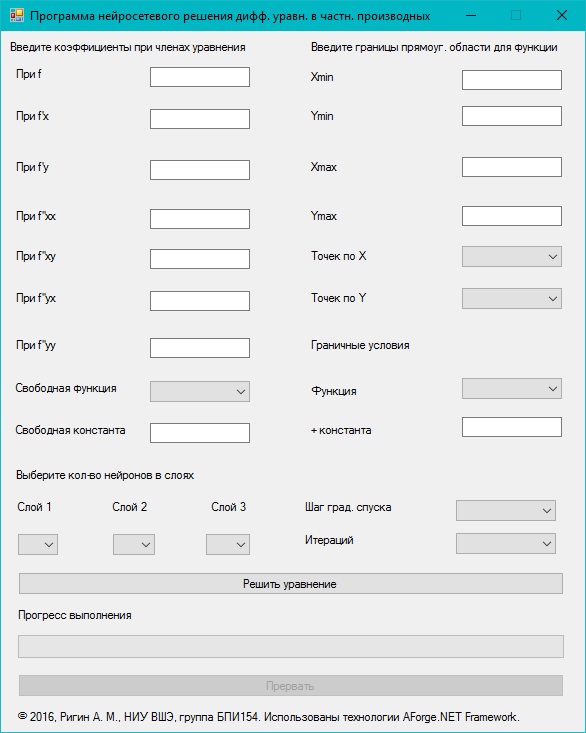
Соответствие программной документации требованиям проверяется путем просмотра программной документации вручную.

Путем просмотра выявлено, что программная документация удовлетворяет требованиям.

## **6.2. Испытание выполнения требований к интерфейсу**

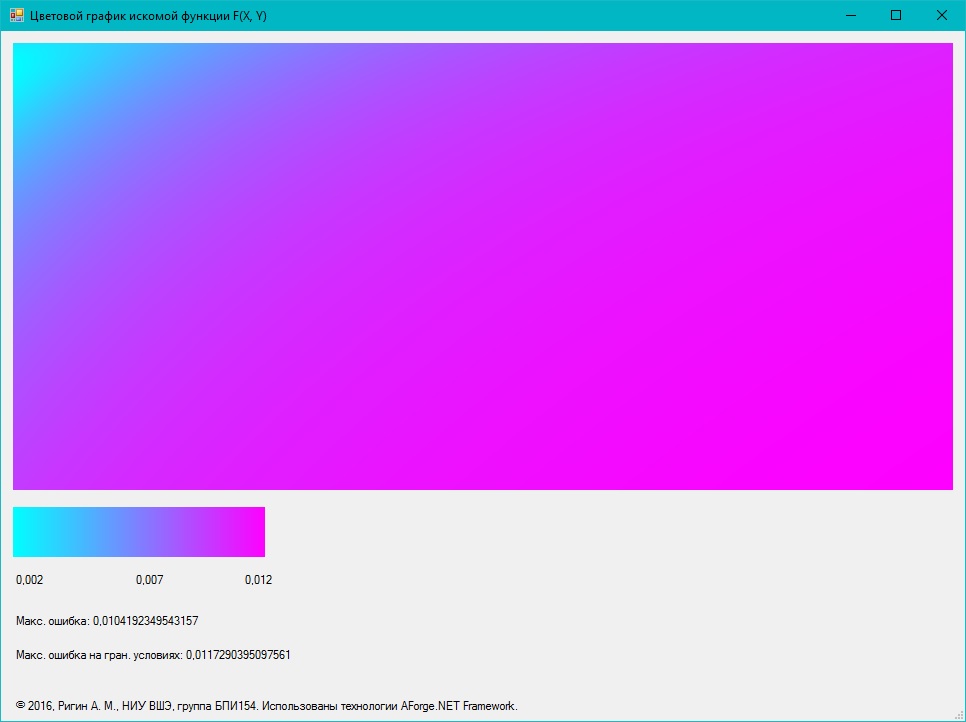
Программа имеет оконный интерфейс Windows Forms.

Ввод данных пользователем осуществляется в главном окне программы:



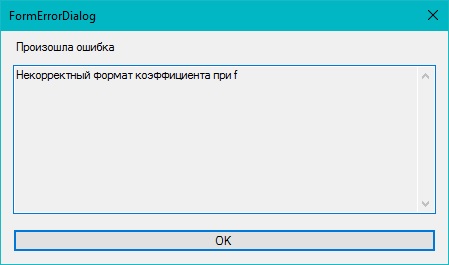
*Рис. 1. Главное окно программы*

Вывод результата осуществляется в окне цветового графика искомой функции:



*Рис. 2. Окно цветового графика искомой функции*

Вывод информации об ошибках производится в специальном диалоговом окне:



*Рис. 3. Диалоговое окно с информацией об ошибке*

Таким образом, программа полностью соответствует требованиям к интерфейсу.

## **6.3. Испытание выполнения требований к функциональным характеристикам**

### **6.3.1. Испытание выполнения требований к функциональным характеристикам в части задания дифференциального уравнения для решения, границ области, на которой дифференциальное уравнение решается, и граничных условий**

Испытание выполнения требований к функциональным характеристикам в части задания дифференциального уравнения для решения, границ области, на которой дифференциальное уравнение решается, и граничных условий выполнено в п. 6.2 настоящего документа, где на рис. 1 показано, что программа обладает всеми вышеперечисленными возможностями.

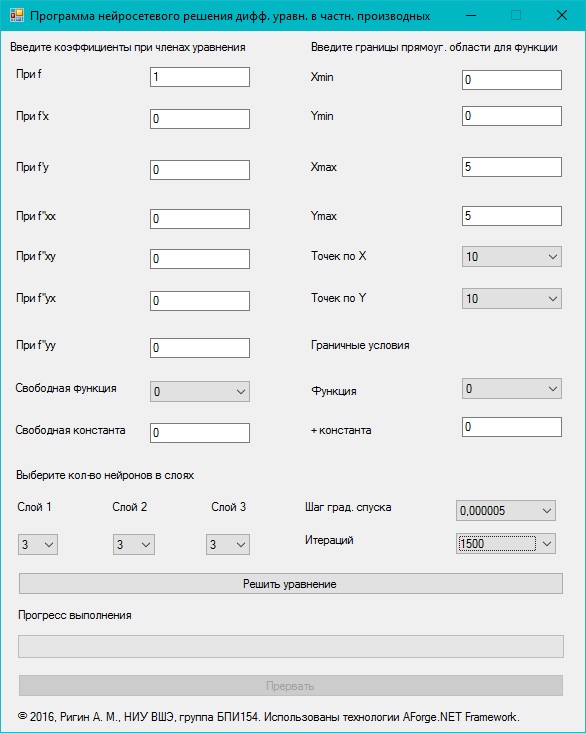
### **6.3.2. Испытание выполнения требований к функциональным характеристикам в части настройки параметров нейронной сети и ее обучения**

Испытание выполнения требований к функциональным характеристикам в части настройки параметров нейронной сети и ее обучения (количество нейронов в слоях нейронной сети, количество итераций обучения, шаг градиентного спуска) выполнено в  
п. 6.2 настоящего документа, где на рис. 1 показано, что программа обладает всеми вышеперечисленными возможностями.

### **6.3.3. Испытание выполнения требований к функциональным характеристикам в части решения заданного дифференциального уравнения нейросетевым способом**

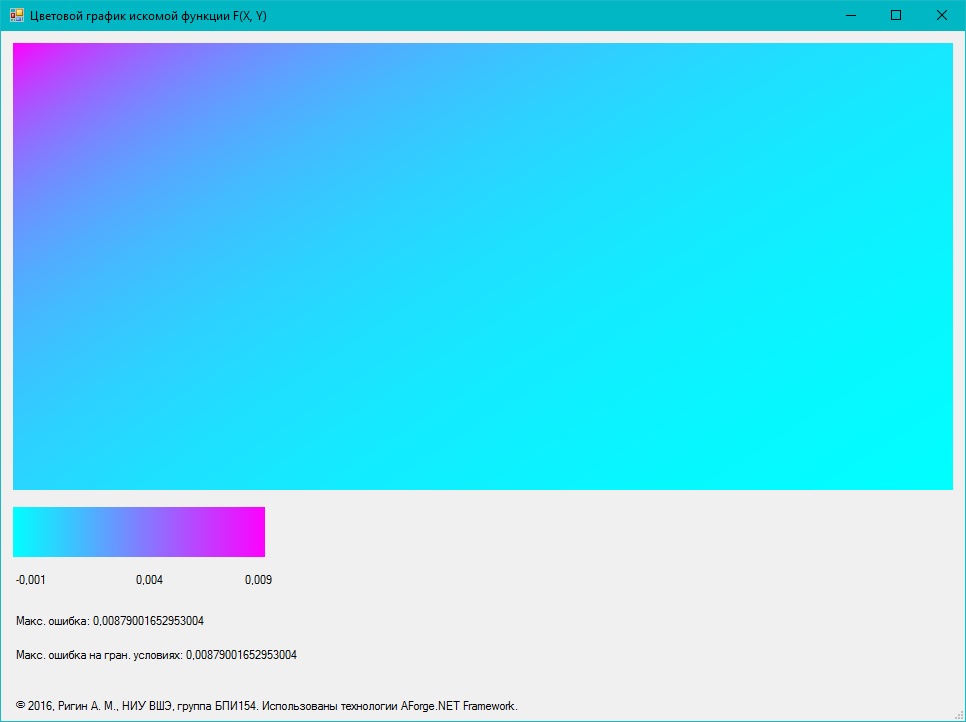
#### **6.3.3.1. Решение дифференциального уравнения f = 0 с граничными условиями f = 0**

Решим дифференциальное уравнение f = 0 с граничными условиями f = 0, задав коэффициенты при f равным 1, при остальных членах дифференциального уравнения равными 0. Свободную функцию и свободную константу установим в значение 0. Выберем следующие границы области: Xmin = 0, Xmax = 0, Ymin = 0, Ymax = 0. Установим количества точек по X и по Y равными 10. Функцию и константу граничных условий установим равными 0. Для решения такого уравнения достаточно 3 нейронов в каждом слое, поэтому установим количества нейронов во всех слоях равными 3. Выберем шаг итерации 0,000005, поскольку он является одним из наиболее оптимальных. Количество итераций установим равным 1500, поскольку 1500 итераций достаточно для решения данного уравнения. Нажмем кнопку «Решить уравнение».



*Рис. 4. Главное окно программы с введенными в него данными для решения дифференциального уравнения f = 0 с граничными условиями f = 0*

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



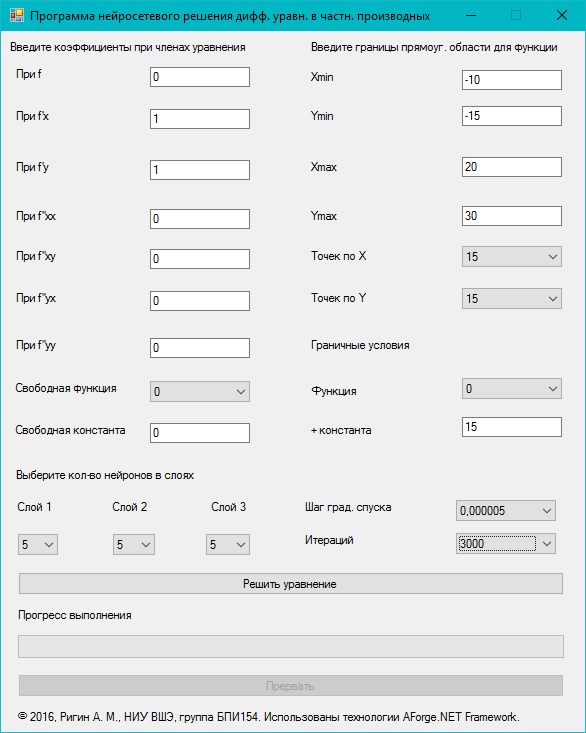
*Рис. 5. Окно с результатом решения дифференциального уравнения f = 0 с граничными условиями f = 0*

Как мы видим, мы с определенной точностью получили функцию, представляющую собой константу, тождественно равную 0, что и в действительности является решением нашего дифференциального уравнения.

Таким образом, программа выполнила поставленную задачу успешно.

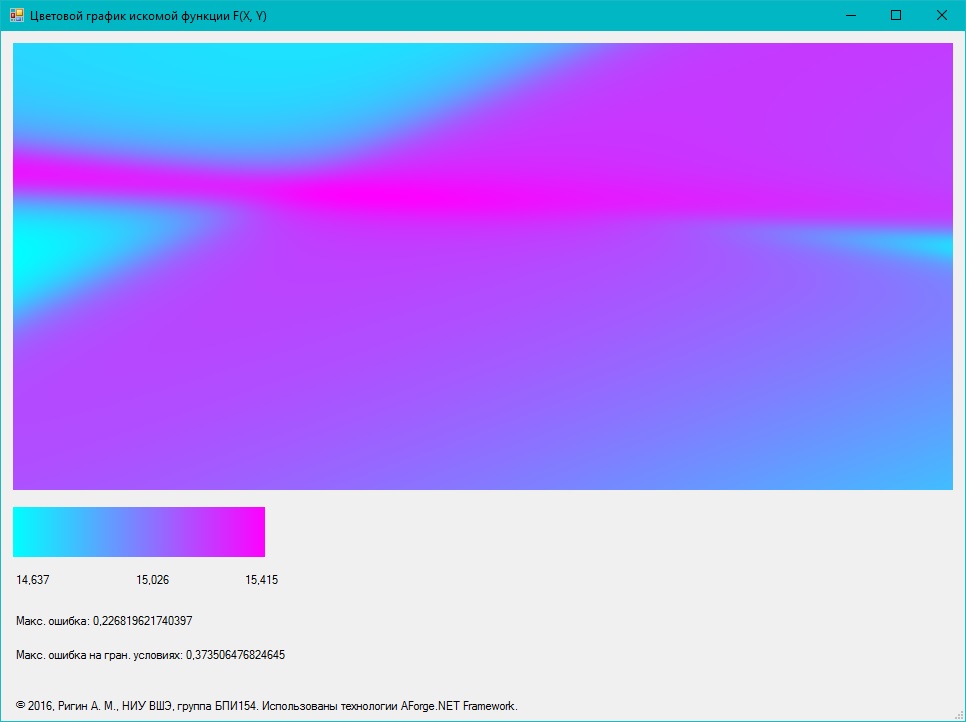
#### **6.3.3.2. Решение дифференциального уравнения f’x + f’y = 0 с граничными условиями f = 15**

Решим дифференциальное уравнение f’x + f’y = 0 с граничными условиями f = 15, задав коэффициенты при f’x и f’y равными 1, при остальных членах дифференциального уравнения равными 0. Свободную функцию и свободную константу установим в значение 0. Выберем следующие границы области: Xmin = -10, Xmax = 20, Ymin = -15, Ymax = 30. Установим количества точек по X и по Y равными 15. Функцию граничных условий установим равной 0, константу граничных условий установим равной 15. Для решения такого уравнения достаточно 5 нейронов в каждом слое, поэтому установим количества нейронов во всех слоях равными 5. Выберем шаг итерации 0,000005, поскольку он является одним из наиболее оптимальных. Количество итераций установим равным 3000, поскольку 3000 итераций достаточно для решения данного уравнения. Нажмем кнопку «Решить уравнение».



*Рис. 6. Главное окно программы с введенными в него данными для решения дифференциального уравнения f’x + f’y = 0 с граничными условиями f = 15*

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



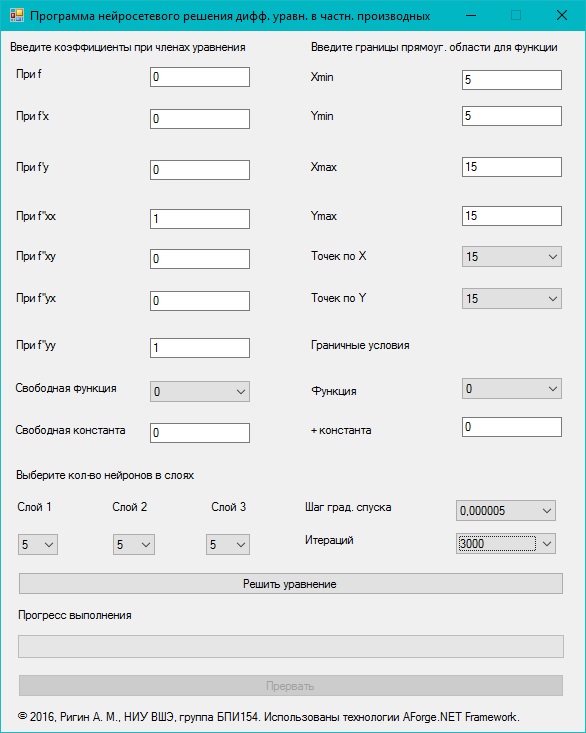
*Рис. 7. Окно с результатом решения дифференциального уравнения f’x + f’y = 0 с граничными условиями f = 15*

Как мы видим, мы с определенной точностью получили функцию, представляющую собой константу, тождественно равную 15, что и в действительности является решением нашего дифференциального уравнения.

Таким образом, программа выполнила поставленную задачу успешно.

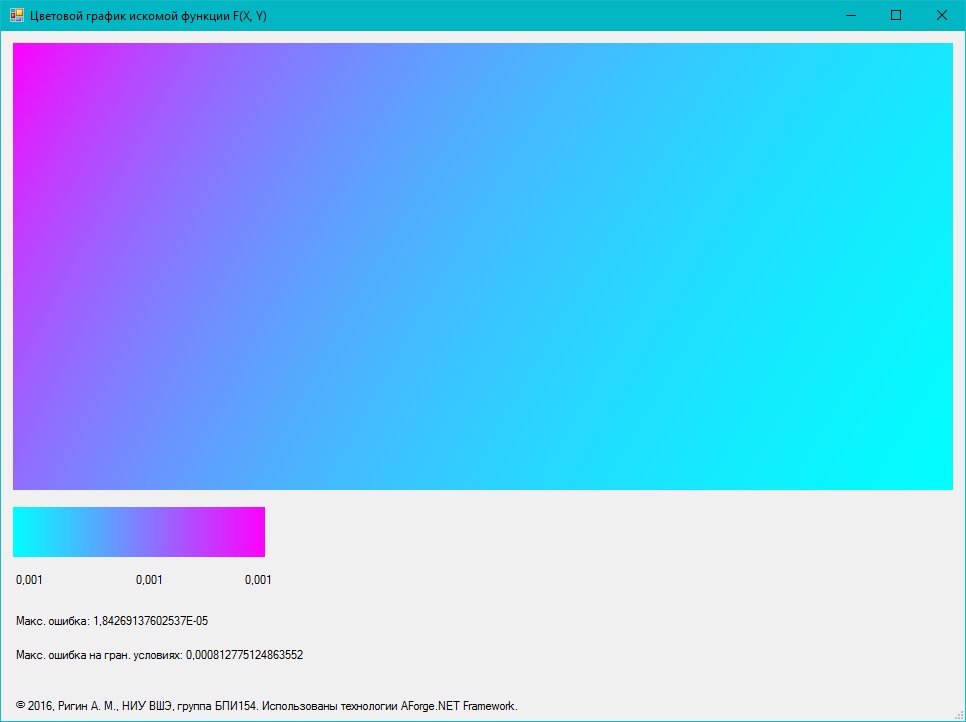
#### **6.3.3.3. Решение дифференциального уравнения f’’xx + f’’yy = 0 с граничными условиями f = 0**

Решим дифференциальное уравнение f’’xx + f’’yy = 0 с граничными условиями  
f = 0, задав коэффициенты при f’’xx и f’’yy равными 1, при остальных членах дифференциального уравнения равными 0. Свободную функцию и свободную константу установим в значение 0. Выберем следующие границы области: Xmin = 5, Xmax = 15,  
Ymin = 5, Ymax = 15. Установим количества точек по X и по Y равными 15. Функцию и константу граничных условий установим равными 0. Для решения такого уравнения достаточно 5 нейронов в каждом слое, поэтому установим количества нейронов во всех слоях равными 5. Выберем шаг итерации 0,000005, поскольку он является одним из наиболее оптимальных. Количество итераций установим равным 3000, поскольку 3000 итераций достаточно для решения данного уравнения. Нажмем кнопку «Решить уравнение».



*Рис. 8. Главное окно программы с введенными в него данными для решения дифференциального уравнения f’'xx + f’’yy = 0 с граничными условиями f = 0*

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



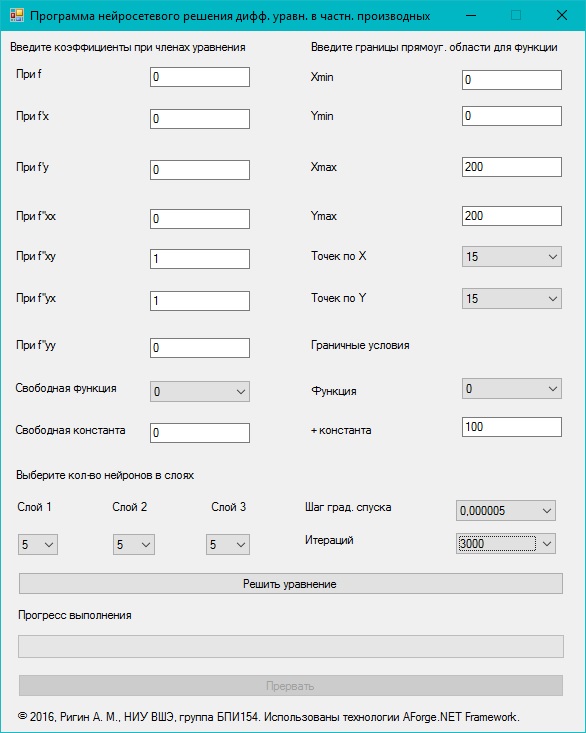
*Рис. 9. Окно с результатом решения дифференциального уравнения f’’xx + f’’yy = 0 с граничными условиями f = 0*

Как мы видим, мы с определенной точностью получили функцию, представляющую собой константу, тождественно равную 0, что и в действительности является решением нашего дифференциального уравнения.

Таким образом, программа выполнила поставленную задачу успешно.

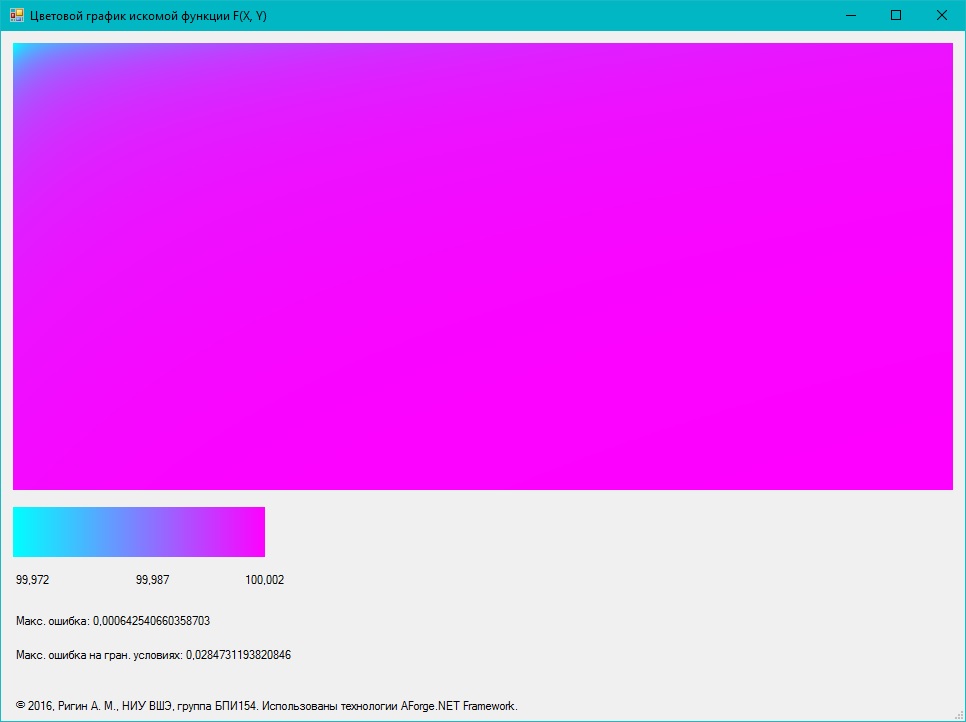
#### **6.3.3.4. Решение дифференциального уравнения f’’xy + f’’yx = 0 с граничными условиями f = 100**

Решим дифференциальное уравнение f’’xy + f’’yx = 0 с граничными условиями  
f = 100, задав коэффициенты при f’’xy и f’’yx равными 1, при остальных членах дифференциального уравнения равными 0. Свободную функцию и свободную константу установим в значение 0. Выберем следующие границы области: Xmin = 0, Xmax = 200,  
Ymin = 0, Ymax = 200. Установим количества точек по X и по Y равными 15. Функцию граничных условий установим равной 0, константу граничных условий установим равной 100. Для решения такого уравнения достаточно 5 нейронов в каждом слое, поэтому установим количества нейронов во всех слоях равными 5. Выберем шаг итерации 0,000005, поскольку он является одним из наиболее оптимальных. Количество итераций установим равным 3000, поскольку 3000 итераций достаточно для решения данного уравнения. Нажмем кнопку «Решить уравнение».



*Рис. 10. Главное окно программы с введенными в него данными для решения дифференциального уравнения f’'xy + f’’yx = 0 с граничными условиями f = 100*

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



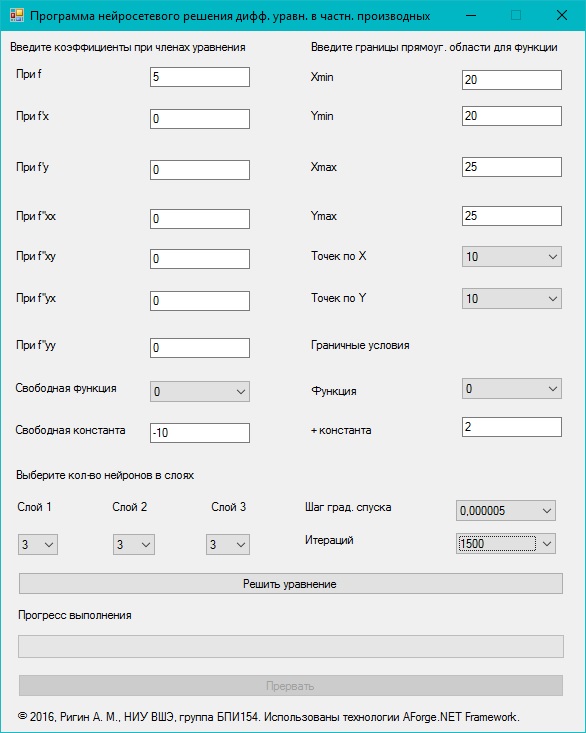
*Рис. 11. Окно с результатом решения дифференциального уравнения f’’xy + f’’yx = 0 с граничными условиями f = 100*

Как мы видим, мы с определенной точностью получили функцию, представляющую собой константу, тождественно равную 100, что и в действительности является решением нашего дифференциального уравнения.

Таким образом, программа выполнила поставленную задачу успешно.

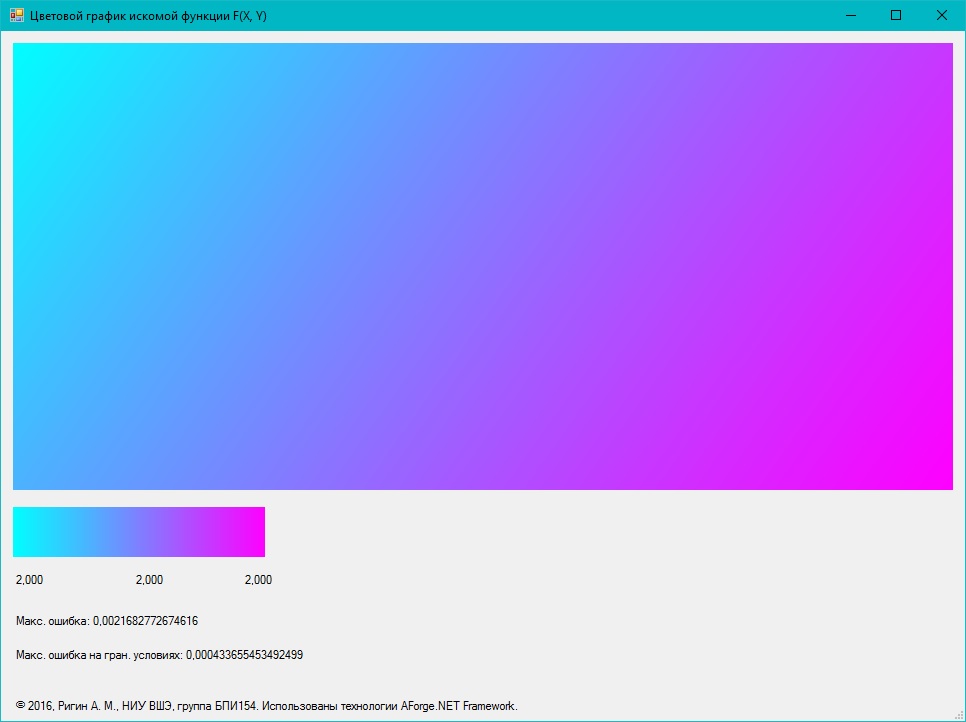
#### **6.3.3.5. Решение дифференциального уравнения 5f - 10 = 0 с граничными условиями f = 2**

Решим дифференциальное уравнение 5f - 10 = 0 с граничными условиями f = 2, задав коэффициенты при f равным 5, при остальных членах дифференциального уравнения равными 0. Свободную функцию установим в значение 0, свободную константу установим в значение -10. Выберем следующие границы области: Xmin = 20, Xmax = 25, Ymin = 20, Ymax = 25. Установим количества точек по X и по Y равными 10. Функцию граничных условий установим равной 0, константу граничных условий установим равной 2. Для решения такого уравнения достаточно 3 нейронов в каждом слое, поэтому установим количества нейронов во всех слоях равными 3. Выберем шаг итерации 0,000005, поскольку он является одним из наиболее оптимальных. Количество итераций установим равным 1500, поскольку 1500 итераций достаточно для решения данного уравнения. Нажмем кнопку «Решить уравнение».



*Рис. 12. Главное окно программы с введенными в него данными для решения дифференциального уравнения 5f - 10 = 0 с граничными условиями f = 2*

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



*Рис. 13. Окно с результатом решения дифференциального уравнения 5f - 10 = 0 с граничными условиями f = 2*

Как мы видим, мы с определенной точностью получили функцию, представляющую собой константу, тождественно равную 2, что и в действительности является решением нашего дифференциального уравнения.

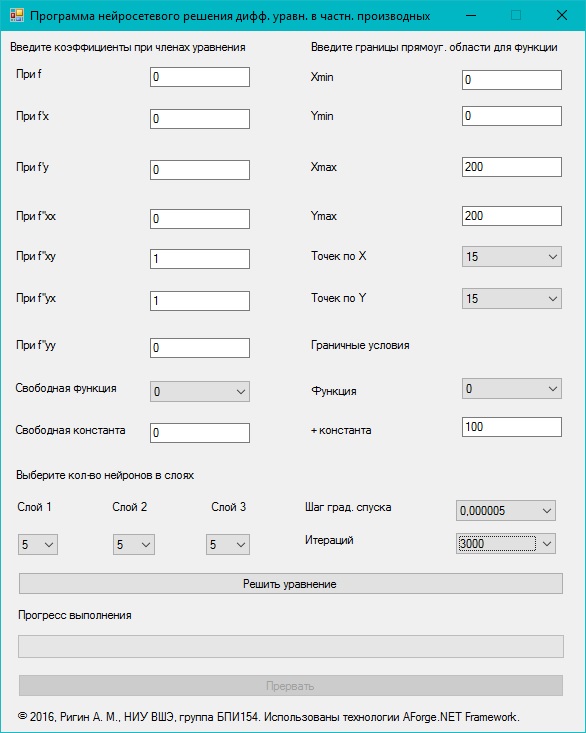
Таким образом, программа выполнила поставленную задачу успешно.

### **6.3.4. Испытание выполнения требований к функциональным характеристикам в части вывода визуализированных результатов решения дифференциального уравнения в форме значений искомой функции на заданной области**

В скриншотах окна с результатами решения дифференциального уравнения в п. 6.3.3 видно, что программа выводит визуализированные результаты решения дифференциального уравнения в форме значений искомой функции на заданной области, а значит, соответствует данному требованию.

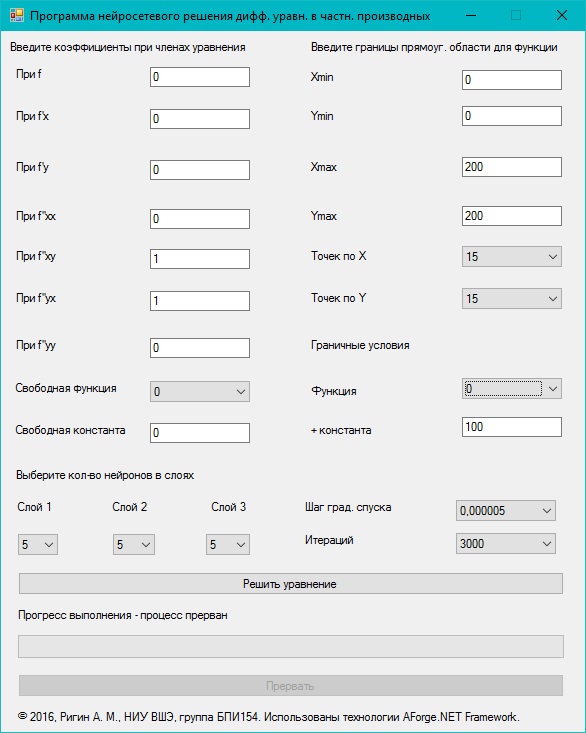
### **6.3.5. Испытание выполнения требований к функциональным характеристикам в части прерывания процесса решения дифференциального уравнения во время его выполнения**

Решим дифференциальное уравнение f’’xy + f’’yx = 0 с граничными условиями  
f = 100, задав коэффициенты при f’’xy и f’’yx равными 1, при остальных членах дифференциального уравнения равными 0. Свободную функцию и свободную константу установим в значение 0. Выберем следующие границы области: Xmin = 0, Xmax = 200,  
Ymin = 0, Ymax = 200. Установим количества точек по X и по Y равными 15. Функцию граничных условий установим равной 0, константу граничных условий установим равной 100. Для решения такого уравнения достаточно 5 нейронов в каждом слое, поэтому установим количества нейронов во всех слоях равными 5. Выберем шаг итерации 0,000005, поскольку он является одним из наиболее оптимальных. Количество итераций установим равным 3000, поскольку 3000 итераций достаточно для решения данного уравнения. Нажмем кнопку «Решить уравнение».



*Рис. 14. Главное окно программы с введенными в него данными для решения дифференциального уравнения f’'xy + f’’yx = 0 с граничными условиями f = 100*

Во время процесса решения дифференциального уравнения нажмем кнопку «Прервать»:



*Рис. 15. Главное окно программы после нажатия кнопки «Прервать»*

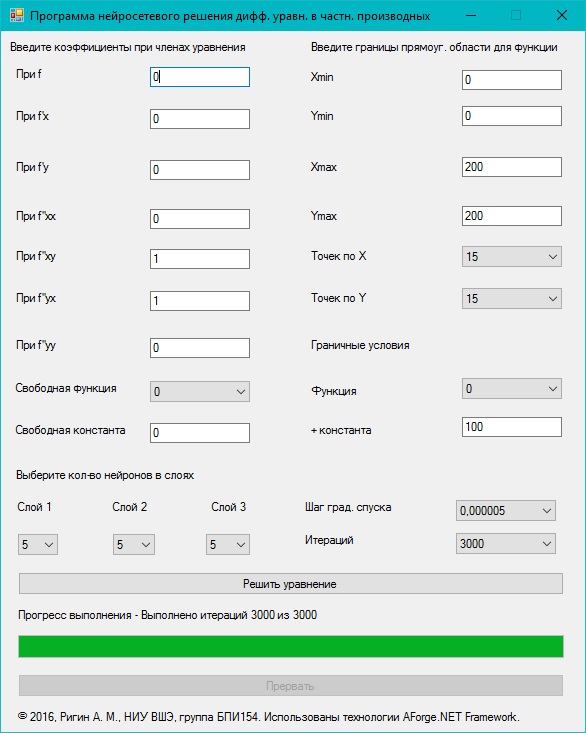
Как мы видим, после нажатия кнопки «Прервать» процесс решения уравнения был прерван, при этом программа сохранила свою работоспособность и готова к решению нового уравнения без перезапуска.

Таким образом, программа соответствует данному требованию.

### **6.3.6. Испытание выполнения требований к функциональным характеристикам в части решения нового дифференциального уравнения без перезапуска программы**

#### **6.3.6.1. Решение нового дифференциального уравнения после завершения решения предыдущего дифференциального уравнения**

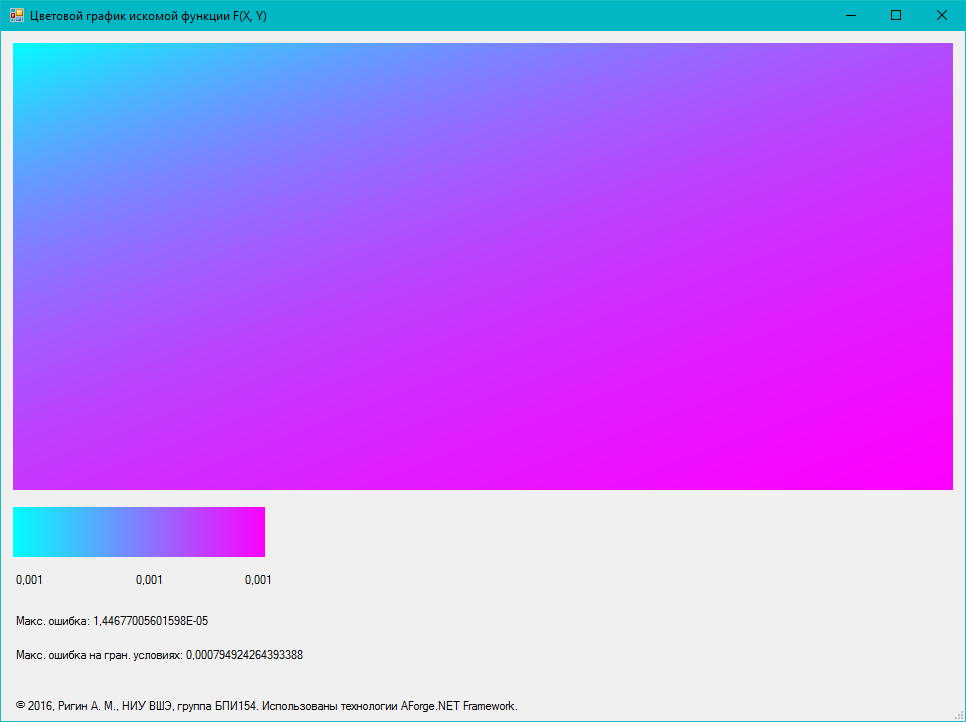
Выполним п. 6.3.3.4 настоящего документа, после чего закроем окно с результатом решения дифференциального уравнения, в результате чего вернемся к главному окну программы.



*Рис. 16. Главное окно программы после закрытия окна с результатами решения дифференциального уравнения*

После этого решим дифференциальное уравнение из п. 6.3.3.3, введя в главном окне соответствующие данные об этом уравнении (указаны в п. 6.3.3.3) и нажав кнопку «Решить уравнение».

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



*Рис. 17. Окно с результатом решения дифференциального уравнения f’’xx + f’’yy = 0 с граничными условиями f = 0*

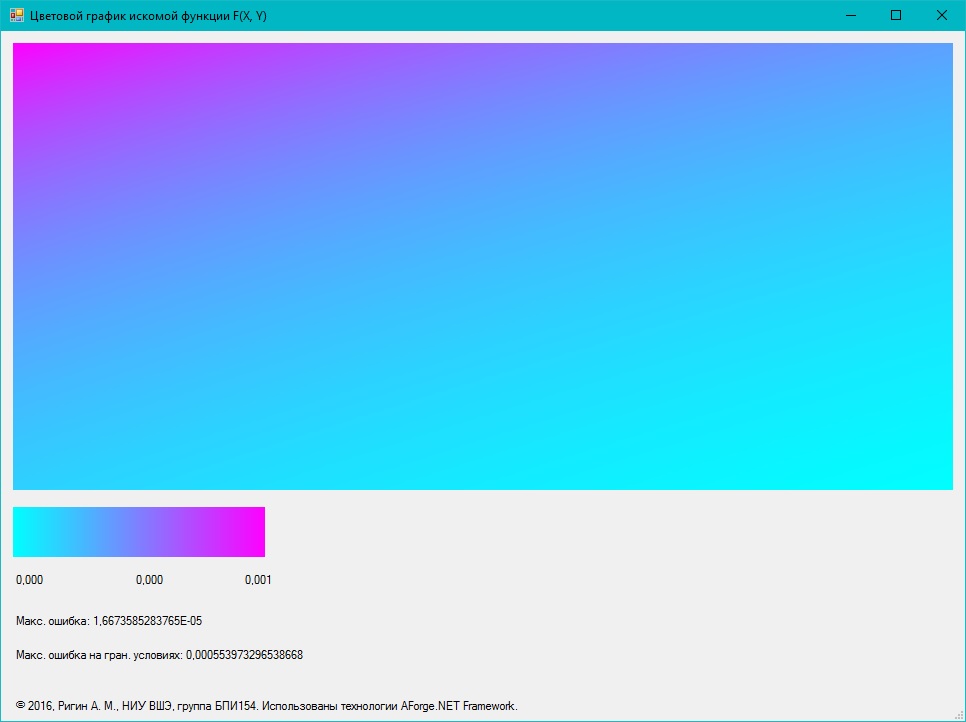
Таким образом, после завершения решения одного дифференциального уравнения мы можем решить новое без перезапуска программы.

#### **6.3.6.2. Решение нового дифференциального уравнения после прерывания решения предыдущего дифференциального уравнения**

Выполним п. 6.3.5 настоящего документа, получим состояние программы, как на рис. 15.

После этого решим дифференциальное уравнение из п. 6.3.3.3 настоящего документа, введя в главном окне соответствующие данные об этом уравнении (указаны в  
п. 6.3.3.3 настоящего документа) и нажав кнопку «Решить уравнение».

После выполнения процесса решения уравнения получим окно с цветовым графиком искомой функции, что и является результатом:



*Рис. 18. Окно с результатом решения дифференциального уравнения f’’xx + f’’yy = 0 с граничными условиями f = 0*

Таким образом, после прерывания решения одного дифференциального уравнения мы можем решить новое без перезапуска программы.

## **6.4. Испытание выполнения требований к надежности**

В пп. 6.2 – 6.3 настоящего документа было обеспечено стабильное и корректное функционирование компьютера и операционной системы. На протяжении всех испытаний в пп. 6.2 – 6.3 настоящего документа программа сохраняла работоспособность.

На рис. 3 в п. 6.2 настоящего документа видно, что программа обеспечивает проверку корректности входных данных.

Таким образом, программа соответствует требованиям к надежности.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Лист регистрации изменений | | | | | | | | | |
| Изм. | Номера листов | | | | Всего листов (страниц) в документе | № документа | Входя- щий № сопро- водит. докум. и дата | Под- пись | Дата |
| изменен- ных | заменен- ных | новых | аннулиро- ванных |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |