

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
вычислительной техники

Кафедра
МОиПЭВМ

Направление подготовки 09.03.04 «Программная инженерия»

БАКАЛАВРСКАЯ РАБОТА

на тему

**Программное обеспечение расчета кратных
интегралов методом статистических
испытаний**

Студент

_____ Дубин Дамир Исхакович
(подпись, дата) (ФИО полностью)

Руководитель

_____ Козлов А.Ю.
(подпись, дата) (фамилия, инициалы)

Нормоконтролёр

_____ Такташкин Д.В.
(подпись, дата) (фамилия, инициалы)

Работа допущена к защите (протокол заседания кафедры от _____ № _____)

Заведующий кафедрой

_____ Козлов А.Ю.
(подпись) (фамилия, инициалы)

Работа защищена с отметкой _____ (протокол заседания ГЭК от _____ № _____)

Секретарь ГЭК

_____ Попова Н.А.
(подпись) (фамилия, инициалы)

Пенза, 2023

Федеральное государственное бюджетное образовательное учреждение
высшего образования
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ

Факультет
Вычислительной техники

Кафедра
МО и ПЭВМ

«Утверждаю»
Заведующий кафедрой
«МО и ПЭВМ»
_____ А.Ю. Козлов
«___» _____ 2023 г

ЗАДАНИЕ
НА ВЫПУСКНУЮ КВАЛИФИКАЦИОННУЮ РАБОТУ
БАКАЛАВРА

1. Студент Дубин Дамир Исакович гр. 19ВП1 факультета ВТ

<Фамилия, имя, отчество полностью>

2. Тема работы Программное обеспечение расчета кратных интегралов методом статистических испытаний

Тема утверждена приказом ПГУ № _____ от “___” _____ 2023 г.

3. Руководитель работы к.т.н., профессор кафедры «МОиПЭВМ» Козлов А.Ю.

4. Задание на работу (назначение разработки, исходные данные и т.п.)

Программное обеспечение расчета кратных интегралов методом статистических испытаний должно быть разработано на языке Matlab; кратность интегралов определяется задачей эффективного воздействия на противодействующую систему; программное обеспечение должно функционировать под управлением ОС Windows 10.0.

5. Перечень подлежащих разработке вопросов

1. Методы расчета кратных интегралов. 1.1. Постановка задачи 1.2. Показатель эффективности воздействия на противодействующую систему. 1.3. Постановка задачи разработки. 1.4. Планирование разработки и оценка бюджета 2. Сущность метода Монте-Карло и моделирование случайных величин. 2.1. Общий алгоритм расчета кратных интегралов методом Монте-Карло. 2.2. Вычисление интегралов. 2.3. Вычисление кратных интегралов 3. Программное обеспечение расчета кратных интегралов методом статистических испытаний при расчете вероятности воздействия на противодействующую систему. 3.1. Характеристика системы ошибок при

воздействия на противодействующую систему. 3.2. Показатель эффективности воздействия на противодействующую систему. 3.3. Алгоритм расчета кратных интегралов методом статистических испытаний Монте-Карло 3.4. Программного обеспечения расчета кратных интегралов методом статистических испытаний

6. Календарный график выполнения работы

№ п/п	Наименование этапов работы	Объем работы	Срок выполнения	Подпись руководителя
1	<i>Анализ предметной области постановка задачи</i>	15%	13.04.2023 – 27.04.2023	
2	<i>Проектирование</i>	25%	27.04.2023 – 10.05.2023	
3	<i>Реализация</i>	40%	10.05.2023 – 30.06.2023	
4	<i>Контроль качества программного обеспечения</i>	5%	31.05.2023 – 02.06.2023	
5	<i>Оформление пояснительной записки</i>	15%	02.06.2023 – 03.06.2023	

Дата выдачи задания "_____" _____ 2023 г.

Руководитель бакалаврской работы _____
(подпись, дата)

Задание к исполнению принял студент _____
(подпись, дата)

Работу к защите допустить
Декан факультета _____
(подпись, дата)

Реферат

Пояснительная записка содержит 89 листов, 14 рисунков, 5 таблиц, 11 использованных источников, 2 приложения.

КРАТНЫЕ ИНТЕГРАЛЫ, МОНТЕ-КАРЛО, MATLAB, UML

Объектом разработки является программное обеспечение для вычисления кратных интегралов.

Цель работы – создание программного обеспечения расчета кратных интегралов методом статистических испытаний на языке Matlab. Программа должна функционировать под управлением ОС Windows 10.0.

Разработка проводилась на языке программирования Matlab в среде разработки MATLAB R2022b.

					ПГУ 09.03.04 – 08ВР191.06 ПЗ			
Изм	Лист	№ докум.	Подп.	Дата	Программное обеспечение расчета кратных интегралов методом статистических испытаний. Пояснительная записка	Лит.	Лист	Листов
Разраб.		Дубин Д.И.					3	89
Пров.		Козлов А.Ю.						
Н. контр.		Такташкин Д.В.				Группа 19ВП1		
Утверд.		Козлов А.Ю.						

Содержание

Введение	6
1 Методы расчета кратных интегралов	7
1.1 Постановка задачи	7
1.2 Численные методы вычисления двойного интеграла	7
1.2.1 Характеристика метода ячеек	7
1.2.2 Последовательное интегрирование с использованием формулы трапеций	14
1.2.3 Последовательное интегрирование с использованием квадратурных формул Гаусса	17
1.2.4 Характеристика метода статистических испытаний (Монте-Карло)	18
1.3 Постановка задачи разработки	24
1.4 Планирование разработки и оценка бюджета	24
2 Сущность метода Монте-Карло и моделирование случайных величин ..	29
2.1 Общий алгоритм расчета кратных интегралов методом Монте-Карло ..	33
2.2 Вычисление интегралов	37
2.3 Вычисление кратных интегралов	41
3 Программное обеспечение расчета кратных интегралов методом статистических испытаний при расчете вероятности воздействия на противодействующую систему	44
3.1 Характеристика системы ошибок при воздействии на противодействующую систему	46
3.2 Показатель эффективности воздействия на противодействующую систему	52
3.3 Алгоритм расчета кратных интегралов методом статистических испытаний Монте-Карло	58
3.4 Программного обеспечения расчета кратных интегралов методом статистических испытаний	65
Заключение	70
Список использованных источников	72

Приложение А. ГЛОССАРИЙ.....	73
Приложение Б. ИСХОДНЫЙ КОД ПРОГРАММЫ.....	75

Введение

Современная методология создания сложных военно-технических систем базируется на всесторонней математизации научных и опытно-конструкторских разработок. Она основана на развитии и широком применении математического моделирования и вычислительного эксперимента. Подход к вычислительным экспериментам был разработан относительно недавно (30-40 лет назад), но уже используется в авиационной, ракетной и космической технике.

Возможность замены исходного объекта его математической «концепцией» и дальнейшего «диалога» с ней таит в себе большие преимущества и означает серьезное изменение методологии и технологии военно-научных исследований. Становится все более ясной неизбежность широкого использования математического моделирования и вычислительного эксперимента для реализации комплексных программ создания сложных военно-технических систем.

В рамках современной методологии исследования сложных систем различают два основных метода моделирования: аналитический и метод статистических испытаний. При этом в недалеком прошлом при исследовании систем они, как правило, взаимно дополняли друг друга. В настоящее время наибольшее распространение в практике моделирования военно-технических систем находит метод статистических испытаний. Этому способствуют два основных фактора: усложнение исследуемых систем и бурное развитие ЭВМ.

Достигнутые успехи в использовании средств вычислительной техники для целей моделирования часто создают иллюзию, что применение современной ЭВМ уже гарантирует возможность исследования системы любой сложности. При этом игнорируется тот факт, что в основу любой модели положено трудоемкое по затратам времени и материальных ресурсов предварительное изучение явлений, имеющих место в объекте-оригинале. Поэтому необходимо постоянно помнить, что ЭВМ – всего лишь инструмент, но не знания.

1 Методы расчета кратных интегралов

Целью данного раздела является изучение численных методов для приближенного вычисления кратных интегралов и их применение.

1.1 Постановка задачи

Пусть требуется вычислить двойной интеграл вида:

$$I = \int \int_D f(x, y) dx dy, \quad (1)$$

где $f(x, y)$ – непрерывная в области D функция двух переменных x и y .

На практике редко удастся выразить интеграл через элементарные функции и найти его точное значение. Поэтому обычно для вычисления интегралов применяются методы численного интегрирования. Они основаны на замене подынтегральной функции $f(x, y)$ аппроксимирующими ее функциями, интегралы от которых легко вычисляются в элементарных функциях. В качестве аппроксимирующих функций, например, можно использовать многочлены.

1.2 Численные методы вычисления двойного интеграла

Рассмотрим наиболее часто используемые способы численного интегрирования: метод ячеек, методы последовательного интегрирования и метод статистического моделирования (метод Монте – Карло).

1.2.1 Характеристика метода ячеек

Пусть сначала область интегрирования является прямоугольником $D = \{a \leq x \leq b, c \leq y \leq d\}$. Среднее значение $\bar{f}(x, y)$ непрерывной в области D функции $f(x, y)$ по теореме о среднем представляется следующим выражением:

$$\bar{f}(x, y) = \frac{1}{S} \int \int_D f(x, y) dx dy, S = (b - a) \cdot (d - c).$$

Считая, что среднее значение приближенно равняется значению функции в центре прямоугольника:

$$\bar{f}(x, y) \approx f(\bar{x}, \bar{y}),$$

где $\bar{x} = (a+b)/2, \bar{y} = (c+d)/2$.

Из соотношения (1) получаем простейшую формулу для приближенного вычисления двойного интеграла:

$$\int \int_D f(x, y) dx dy \approx S f(\bar{x}, \bar{y}). \quad (2)$$

Найдем погрешность формулы (2). Функцию $f(x, y)$ будем считать достаточно гладкой, т. е. будем полагать, что она имеет все необходимые по ходу рассуждения непрерывные производные. Разложим функцию $f(x, y)$ по формуле Тейлора, выбирая центр прямоугольника (точку $(\bar{x}, \bar{y})$) за точку разложения

$$\begin{aligned} f(x, y) = & f(\bar{x}, \bar{y}) + (x - \bar{x}) f'_x(\bar{x}, \bar{y}) + (y - \bar{y}) f'_y(\bar{x}, \bar{y}) + \\ & + \frac{1}{2} (x - \bar{x})^2 f''_{xx}(\bar{x}, \bar{y}) + (x - \bar{x})(y - \bar{y}) f''_{xy}(\bar{x}, \bar{y}) + \\ & + \frac{1}{2} (y - \bar{y})^2 f''_{yy}(\bar{x}, \bar{y}) + \dots \end{aligned} \quad (3)$$

Погрешность есть разность точного и приближенного значений интеграла.

Подставляя в (2) формулу (3), получим главный член погрешности

$$\begin{aligned} R = \int_a^b \int_c^d f(x, y) dx dy - S f(\bar{x}, \bar{y}) \approx \\ \approx \frac{1}{24} S \left[(b - a)^2 f''_{xx}(\bar{x}, \bar{y}) + (d - c)^2 f''_{yy}(\bar{x}, \bar{y}) \right], \end{aligned} \quad (4)$$

где члены, отброшенные при замене точного равенства приближенным, содержат производные старших порядков и более высокие степени длин сторон прямоугольника D . Заметим, что все члены разложения, являющиеся нечетными функциями относительно центра прямоугольника, не вносят вклад в погрешность, поскольку интегралы от этих членов оказываются равными нулю.

В общем случае длины сторон прямоугольника $(b - a)$ и $(d - c)$ не малы, поэтому главный член погрешности (4) может быть велик. Для повышения

точности вычислений в области D (рисунок 1) вводится сетка $x_i = a + ih_1$, $y_j = a + jh_2$, $i=0,1,\dots,m$, $j=0,1,\dots,n$; $h_1 = (b-a)/m$, $h_2 = (c-d)/n$, с достаточно мелкими ячейками $\square D_{ij} = \{x_{i-1} \leq x \leq x_i, y_{j-1} \leq y \leq y_j, i=1,\dots,m; j=1,\dots,n\}$.

Вычисляя интеграл по каждой ячейке по формуле (2) и суммируя найденные значения по всем ячейкам, получаем формулу метода ячеек

$$I \approx I_h = \sum_{i=1}^m \sum_{j=1}^n S_{ij} \cdot f(\bar{x}_i, \bar{y}_j), \quad (5)$$

где $S_{ij} = h_1 \cdot h_2$ – площадь ячейки;

$\bar{x}_i = (x_{i-1} + x_i)/2$, $\bar{y}_j = (y_{j-1} + y_j)/2$ – координаты центра ячейки.

Здесь и далее пусть I_h будет приближенным значением интеграла (1), вычисленное по формуле (5) с шагами h_1 и h_2 .

Справа в выражении (5) стоит интегральная сумма, поэтому для любой непрерывной функции $f(x, y)$ эта сумма сходится к значению интеграла, когда периметры всех ячеек стремятся к нулю.

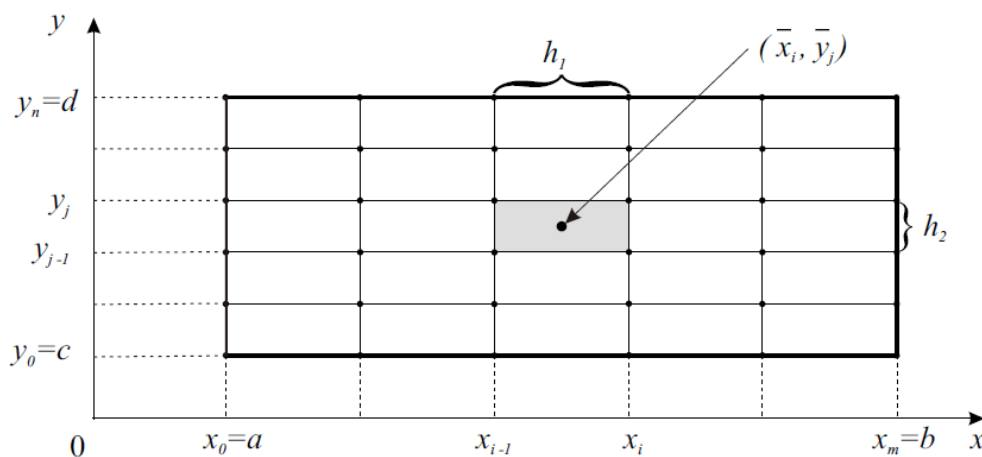


Рисунок 1 – Приближенное вычисление двойного интеграла

Погрешность интегрирования (4) для одной ячейки $\square D_{ij}$ представляется в виде:

$$R_{ij} \approx \frac{1}{24} S_{ij} \left[h_1^2 \cdot f''_{xx}(\bar{x}_i, \bar{y}_j) + h_2^2 \cdot f''_{yy}(\bar{x}_i, \bar{y}_j) \right]. \quad (6)$$

Суммируя выражения (6) по всем ячейкам, получаем погрешность метода ячеек:

$$R \approx c_1 \cdot h_1^2 + c_2 \cdot h_2^2, \quad (7)$$

где

$$c_1 = \frac{1}{24} \int_D \int f_{xx}''(x, y) dx dy, \quad c_2 = \frac{1}{24} \int_D \int f_{yy}''(x, y) dx dy,$$

или

$$R = O(h_1^2 + h_2^2), \quad (8)$$

т. е. метод ячеек имеет второй порядок точности относительно шагов сетки h_1 и h_2 .

Заметим, что поскольку в оценке (6) отброшены более высокие степени h_1 и h_2 , то соотношение для погрешности (7) является асимптотическим, т. е. выполняется при $h_1 \rightarrow 0$ и $h_2 \rightarrow 0$ с точностью до членов более высокого порядка малости по h_1 и h_2 .

Для вычисления интеграла (1) с заданной точностью можно воспользоваться, как это следует из (8), правилом Рунге практической оценки погрешности. С помощью разложения $f(x, y)$ в окрестности центра каждой ячейки по формуле Тейлора до членов с производными четвертого порядка можно получить оценку не только главного члена погрешности (8), но и оценить следующие по порядку малости h_1 и h_2 члены погрешности. Заметим, что члены в разложении $f(x, y)$ по формуле Тейлора, содержащие производные третьего порядка, в силу симметрии области интегрирования ΔD_{ij} относительно точки разложения не вносят вклад в погрешность интегрирования. Поэтому, для того чтобы учесть следующие после главного члена по порядку малости h_1 и h_2 члены погрешности, необходимо разлагать $f(x, y)$ до членов, содержащих производные четвертого порядка. В результате интеграл (1) можно представить в виде:

$$I = I_h + c_1 + h_1^2 + c_2 + h_2^2 + O(h_1^4 + h_1^2 \cdot h_2^2 + h_2^4). \quad (9)$$

Выражения для c_1 и c_2 были приведены выше (см. формулу (7)). Здесь важно подчеркнуть, что c_1 и c_2 – не зависящие от h_1 и h_2 постоянные величины, причем они не должны одновременно обращаться в 0.

Таким образом, если известен главный член погрешности, то можно увеличить точность вычисления интеграла (1):

$$I \approx I_h + c_1 + h_1^2 + c_2 + h_2^2. \quad (10)$$

Однако постоянные c_1 и c_2 являются неизвестными величинами. Для того, чтобы вычислить интеграл (1) с учетом главного члена погрешности, можно поступить следующим образом. Сначала находим значение I_h , затем – значение $I_{h/2}$. Здесь $I_{h/2}$ – значение интеграла (1), вычисленное по формуле (5) с шагами $h_1/2$ и $h_2/2$. Теперь, наряду с выражением (10), можно написать следующее соотношение:

$$I \approx I_{h/2} + c_1 \cdot \left(\frac{h_1}{2}\right)^2 + c_2 \cdot \left(\frac{h_2}{2}\right)^2. \quad (11)$$

Тот факт, что сетка по каждой переменной x и y сгущается в одинаковое число раз, позволяет в выражении (10) выделить главный член погрешности $[c_1 \times (h_1/2)^2 + c_2 \times (h_2/2)^2]$ формулы (11):

$$I \approx I_h + 4[c_1 \cdot \left(\frac{h_1}{2}\right)^2 + c_2 \cdot \left(\frac{h_2}{2}\right)^2] \quad (12)$$

Из выражений (11) и (12) следует,

$$I_h + 4[c_1 \cdot \left(\frac{h_1}{2}\right)^2 + c_2 \cdot \left(\frac{h_2}{2}\right)^2] \approx I_{h/2} + c_1 \cdot \left(\frac{h_1}{2}\right)^2 + c_2 \cdot \left(\frac{h_2}{2}\right)^2.$$

Из этого соотношения получаем выражение для главного члена погрешности формулы (11):

$$c_1 \cdot \left(\frac{h_1}{2}\right)^2 + c_2 \cdot \left(\frac{h_2}{2}\right)^2 \approx \frac{I_{h/2} - I_h}{3}. \quad (13)$$

Теперь, согласно формуле (11), имеем приближенную оценку погрешности по правилу Рунге:

$$I - I_{h/2} \approx \frac{I_{h/2} - I_h}{3}. \quad (14)$$

Наконец, подставляя выражение (13) в (11), получаем значение интеграла (1) с учетом главного члена погрешности,

$$I \approx I_h^* = \frac{4 \cdot I_{h/2} - I_h}{3},$$

где I_h^* – уточненное по Ричардсону значение интеграла I .

Подчеркнем, что для практической оценки погрешности по правилу Рунге сетка по каждой переменной сгущается в одинаковое число раз, т. е. отношение m/n при сгущении сетки должно оставаться постоянным. В противном случае не удастся в результате двойного пересчета интеграла (1) по двум сеткам с разными размерами ячеек составить формулы типа (10) и (11), из которых можно найти главный член погрешности.

Понятно, что если одновременно $c_1 = 0$ и $c_2 = 0$, то для оценки погрешности вычисления интеграла (1) правило Рунге в виде (14) неприменимо.

Мы получили формулу (5) для вычисления интеграла в простейшем случае – для прямоугольной области. Если область не прямоугольная, то в ряде случаев исходный интеграл по такой области соответствующей заменой переменных удобно преобразовать к двойному интегралу по прямоугольной области. Например, если область задана в виде криволинейного четырехугольника $D = \{a \leq x \leq b, \varphi_1(x) \leq y \leq \varphi_2(x)\}$ (рисунок 2, а), то с помощью замены переменных $x = x(u) = (b - a)u + a$, $y = \varphi_1(x(u)) + v \times (\varphi_2(x(u)) - \varphi_1(x(u)))$ исходная область D преобразуется в квадратную область $D' = \{0 \leq u \leq 1, 0 \leq v \leq 1\}$ (рисунок 2, б).

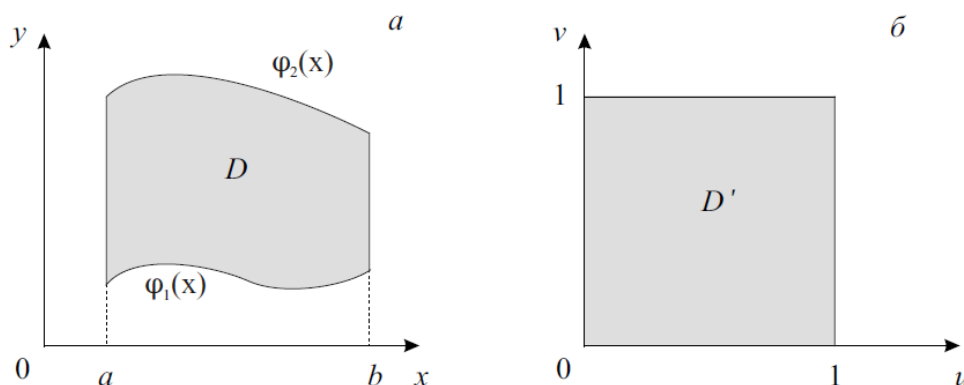


Рисунок 2 – Области четырехугольников D и D'

Напомним правило замены переменных в двойном интеграле. Если ограниченная замкнутая область D в плоскости O_{xy} взаимно однозначно отображается на область D' на плоскости O_{uv} с помощью непрерывно дифференцируемых функций $x = x(u, v)$, $y = y(u, v)$, причем якобиан преобразования:

$$J = \begin{vmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{vmatrix} \neq 0,$$

то справедлива формула:

$$\int \int_D f(x, y) dx dy = \int \int_{D'} f(x(u, v), y(u, v)) |J| du dv.$$

Методом ячеек можно вычислить интеграл и по области сложной формы, например, с криволинейной границей (рисунок 3).

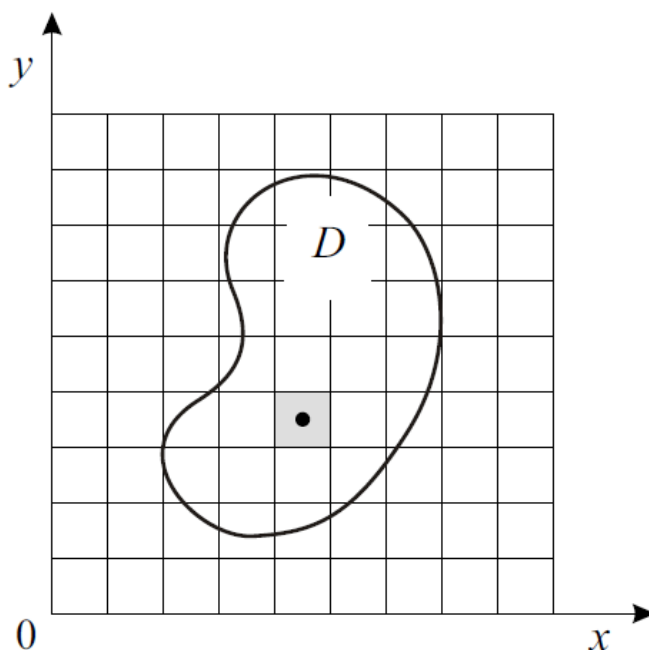


Рисунок 3 – График криволинейной границы со случайной точкой

Интеграл в этом случае будем вычислять следующим образом. Наложим на область D прямоугольную сетку, и в интегральную сумму (5) будем включать только те ячейки, все точки которых принадлежат области D . В итоге на порядок понижается точность формулы (5), поэтому для вычисления интеграла с достаточной точностью требуется сетка с более мелкими ячейками.

Следует отметить, что метод ячеек (5) легко переносится на большее число измерений (для вычисления тройных и большей кратности интегралов). В случае однократного интеграла аналогом метода ячеек является метод средних прямоугольников.

1.2.2 Последовательное интегрирование с использованием формулы трапеций

Другой метод вычисления двойных интегралов – их сведение к последовательному вычислению однократных интегралов. Снова рассмотрим интеграл по прямоугольной области $D = \{a \leq x \leq b, c \leq y \leq d\}$ (рисунок 4). Интеграл (1) можно вычислить последовательным интегрированием:

$$I = \int \int_D f(x, y) dx dy = \int_c^d \left(\int_a^b f(x, y) dx \right) dy. \quad (15)$$

Это выражение перепишем в следующий вид:

$$I = \int_c^d F(y) dy, \quad F(y) = \int_a^b f(x, y) dx. \quad (16)$$

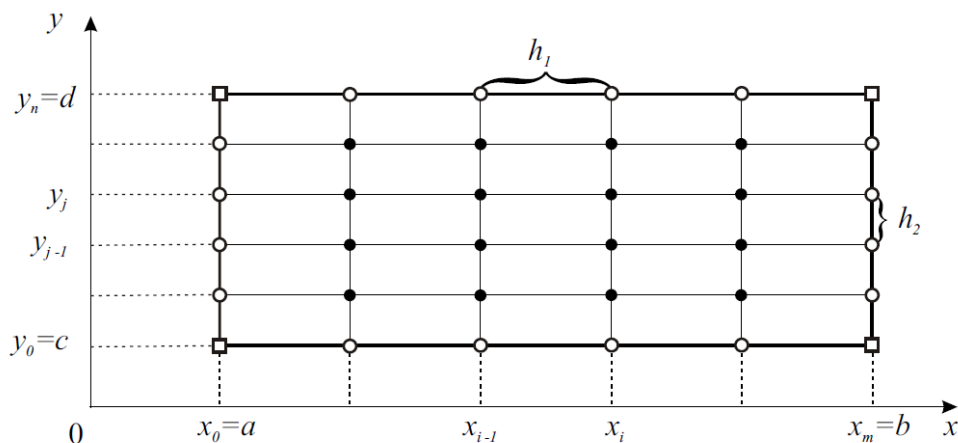


Рисунок 4 – График интеграла по прямоугольной области

Для вычисления этих интегралов могут быть использованы известные формулы из численного метода вычисления определенного интеграла. Например, пусть и по направлению x , и по направлению y для приближенного вычисления применяется формула трапеций.

Из этого следует:

$$F(y_j) \approx h_1 \cdot \sum_{i=0}^m q_{1,i} \cdot f(x_i, y_j), \quad (17)$$

где

$$q_{1,i} = \begin{cases} 1/2 & \text{при } i=0 \text{ и } i=m, \\ 1 & \text{при } i=1, 2, \dots, m-1. \end{cases}$$

и

$$I \approx h_2 \cdot \sum_{j=0}^m q_{2,j} \cdot F(y_j), \quad (18)$$

где

$$q_{2,j} = \begin{cases} 1/2 & \text{при } j=0 \text{ и } i=n, \\ 1 & \text{при } j=1,2,\dots,n-1. \end{cases}$$

Подставляя выражение (17) в (18), получаем формулу последовательного интегрирования:

$$I \approx I_h = h_1 \cdot h_2 \cdot \sum_{i=0}^m \sum_{j=0}^n q_{i,i} \cdot f(x_i, y_j), \quad (19)$$

где

$$q_{ij} = q_{1,i} \cdot q_{2,j} = \begin{cases} 1/4 & \text{при } i=0,m; \quad j=0,n; \\ 1/2 & \text{при } \begin{cases} i=0 & \text{и } i=m & j=1,\dots,n-1, \\ j=0 & \text{и } j=n & i=1,\dots,m-1; \end{cases} \\ 1 & \text{при } i=1,\dots,m-1, \quad j=1,\dots,n-1. \end{cases}$$

На рисунке 4 приведена сетка, которая используется при приближенном вычислении интеграла (1) по формуле (19). Точками, кружочками и квадратиками показаны узловые точки, в которых коэффициенты $q_{ij} = 1, 1/2$ и $1/4$, соответственно.

Легко убедиться в том, что для дважды непрерывно дифференцируемой функции $f(x, y)$ формула (19) имеет второй порядок точности относительно шагов h_1 и h_2 и что можно применить правило Рунге практической оценки погрешности.

Если в методе последовательного интегрирования воспользуемся формулой средних прямоугольников при интегрировании по каждому из направлений x и y , то в результате получим расчетную формулу метода ячеек (5).

Далее рассмотрим случай сложной области. Метод последовательного интегрирования можно непосредственно применять и к области произвольной формы, например, с криволинейной границей (см. рисунок 3). Однако для получения простых расчетных формул на практике всегда стараются свести исходный интеграл к сумме интегралов по прямоугольным областям.

1.2.3 Последовательное интегрирование с использованием квадратурных формул Гаусса

Для получения квадратурной формулы более высокой точности, чем формулы (19) можно воспользоваться квадратурными формулами Гаусса. При этом предварительно заменой переменных $x(u) = (a+b)/2 + u \cdot (b-a)/2$, $y(v) = (c+d)/2 + v \cdot (d-c)/2$ прямоугольная область $\{a \leq x \leq b, c \leq y \leq d\}$ преобразуется в квадратную область $D = \{-1 \leq u \leq 1, -1 \leq v \leq 1\}$. Поэтому будем считать, что с самого начала требуется вычислить интеграл по области $D = \{-1 \leq x \leq 1, -1 \leq y \leq 1\}$:

$$I = \int \int_D f(x, y) dx dy = \int_{-1}^1 \int_{-1}^1 f(x, y) dx dy. \quad (20)$$

Применяя для интегрирования (20) и по направлению x , и по направлению y квадратурную формулу Гаусса с одинаковым числом узлов, получаем следующую формулу последовательного интегрирования:

$$I_n = \sum_{i=1}^n \sum_{j=1}^n q_i \cdot q_j \cdot f(x_i, y_j). \quad (21)$$

Значения координат узловых точек и весовых коэффициентов по направлениям x и y берутся из таблицы 1.1. Расположение узловых точек для $n = 3$ и $n = 4$ проиллюстрировано на рисунке 5.

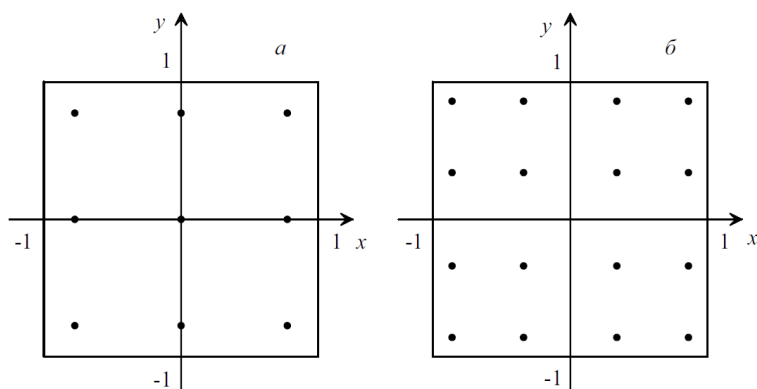


Рисунок 5 – Графики узловых точек

Таблица 1 – Координаты узловых точек и весовые коэффициенты
квadrатурной формулы Гаусса

Число узлов n	Номер точки i	Координаты точки x_i	Коэффициент q_i
1	1	0	1
2	1	$x_1 = -x_2$	1
	2	0.5773502692	1
3	1	$x_1 = -x_3$	$q_1 = q_3$
	2	0	0.8888888889
	3	0.7745966692	0.5555555556
4	1	$x_1 = -x_2$	$q_1 = q_4$
	2	$x_1 = -x_2$	$q_2 = q_3$
	3	0.3399810436	0.6521451549
	4	0.8611363116	0.3478548451
5	1	$x_1 = -x_2$	$q_1 = q_5$
	2	$x_1 = -x_2$	$q_2 = q_3$
	3	0	0.5688888889
	4	0.5384693101	0.4786286705
	5	0.9061798459	0.2369268851

1.2.4 Характеристика метода статистических испытаний (Монте-Карло)

При расчёте интегралов методом Монте-Карло реализуется следующая основная идея.

Основные вероятностные характеристики непрерывной случайной величины x определяются интегралами, связанными с её п. р. в. $f(t)$. К их числу относятся следующие характеристики:

– вероятность попадания x в интервал (a,b) :

$$P\{a \leq x \leq b\} = \int_a^b f(t)dt; \quad (22)$$

– математическое ожидание x :

$$M(x) = \int_{-\infty}^{\infty} t \cdot f(t)dt; \quad (23)$$

– математическое ожидание любой функции φ от с. в. x :

$$M[\varphi(x)] = \int_{-\infty}^{\infty} \varphi(t) \cdot f(t)dt; \quad (24)$$

– математическое ожидание x^r (r -й начальный момент):

$$M(x^r) = \int_{-\infty}^{\infty} t^r \cdot f(t)dt. \quad (25)$$

Часто эти интегралы сложны для вычисления. При статистическом моделировании вместо точных значений (22) – (25) можно вычислять и использовать соответствующие приближённые эмпирические оценки, легко рассчитываемые по выборке $x_1, x_2, \dots, x_n$:

– оценку вероятности попадания x в интервал (a, b) :

$$P\{a \leq x \leq b\} = \frac{1}{n} \sum_{i=1}^n I[a \leq x_i \leq b], \quad (26)$$

где функция $I[A]$ – индикатор события A – равна 1, когда событие A имеет место, и равна нулю в противном случае;

– оценку математического ожидания x :

$$M(x) = \frac{1}{n} \sum_{i=1}^n x_i; \quad (27)$$

– оценку математического ожидания любой функции $\varphi(x)$ от с. в. x :

$$M[\varphi(x)] = \frac{1}{n} \sum_{i=1}^n \varphi(x_i), \quad (28)$$

и, в частности, оценку математического ожидания x^r (r -го начального момента):

$$M(x^r) = \frac{1}{n} \sum_{i=1}^n x_i^r. \quad (29)$$

Оценки (26) – (29) легко вычисляются, и погрешности, с которыми они приближают точные значения характеристик, легко контролируются.

Действительно, как суммы большого числа независимых с. в. они имеют нормальное распределение. При этом м. о. оценок характеристик совпадают с точными значениями характеристик, а фактические отклонения оценок от точных значений характеристик подчиняются известному правилу «трёх сигм» [1] и могут быть сделаны сколь угодно малыми путём увеличения объёма выборки n . Более подробные сведения о статистических оценках, включая теорию состоятельности, несмещённости и эффективности оценок, можно найти в учебниках по математической статистике [2, 3]. Основная идея метода Монте-Карло состоит в том, чтобы использовать расчёт эмпирических оценок в качестве способа вычисления определённых интегралов вида (22) – (25). Поскольку к такому виду легко сводятся практически любые определённые интегралы, метод Монте-Карло оказывается достаточно универсальным методом. Преимущества метода Монте-Карло и условия, в которых они проявляются, можно увидеть, рассмотрев некоторые примеры его применения.

Сведение интеграла к вероятности.

Наиболее простой способ вычисления определённого интеграла методом Монте-Карло – это сведение интеграла к вероятности. Суть этого способа легко понять из следующего примера.

Рассчитаем методом Монте-Карло определённый интеграл:

$$S = \int_0^1 \sqrt{1-t^2} dt, \quad (30)$$

который представляет собой площадь четверти круга единичного радиуса с центром в начале координат (рисунок 6).

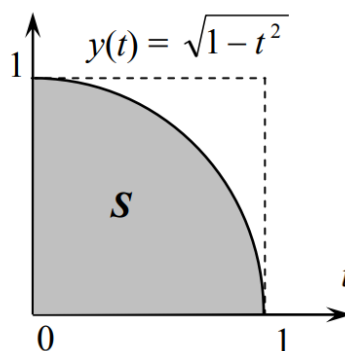


Рисунок 6 – График для расчета интеграла

Если в единичный квадрат, ограниченный на рисунке 6 координатными осями и штриховой линией, бросить случайную точку, то она попадёт в круг с вероятностью $p = S/S_0$, где S_0 – площадь квадрата известна. Отсюда следует, что искомое значение интеграла $S = pS_0$ однозначно выражается через вероятность p . В данном случае $S_0 = 1$, значит $S = p$. Вероятность p легко можно оценить методом Монте-Карло, выбрасывая в квадрат достаточно большое число n случайных точек. Оценка p вероятности p вычисляется при этом по формуле

$$p = \frac{1}{n} \sum_{i=1}^n I(A_i), \quad (31)$$

где событие A_i – это событие попадания точки в круг.

Случайные точки должны иметь равномерное распределение внутри квадрата. Формируются выборки в количестве 10000 штук путем программной генерации базовых случайных величин. Далее производится оценка вероятности по формуле (31) следующим образом. Вычисляется индикатор попадания точки в единичный круг. Индикатору присваивается значение 1, если сумма квадратов координат случайной точки не превосходит 1, и значение 0 – в противном случае. На этом завершается расчёт приближённого значения искомого определённого интеграла S методом Монте-Карло.

Зная точное значение величины S , составляющее, очевидно, $S = \pi/4 = 0,785398$, вычисляют фактическое отклонение приближённой оценки от точной. Нетрудно убедиться, что погрешность не выходит за пределы, определённые правилом «трёх сигм». Согласно этому правилу, с вероятностью $1-0,0027$ погрешность Δ эмпирической оценки лежит в пределах $M(x)$,

$$|\Delta| = |M(x) - M(x)| \leq 3\sigma_n, \quad (32)$$

где $\sigma_n = \sqrt{D(x)/n}$ – среднее квадратичное отклонение оценки $M(x)$;

$D(x)$ – дисперсия усредняемой выборочной с. в. x ;

n – объём выборки.

Формула (32), опирающаяся на значение дисперсии $D(x)$ усредняемой с. в. x , является основной формулой планирования статистических экспериментов. При вычислении эмпирической вероятности усредняемая с. в. x является индикатором того случайного события, вероятность которого нужно оценить, поэтому $x \in \{0, 1\}$. Легко установить, что дисперсия $D(x) = p(1-p)$, где $p = P\{x = 1\}$ – вероятность индицируемого события. В нашем случае p известна: $p = S = \pi/4 = 0,785398$, откуда $D(x) = p(1-p) = 0,168548$. Поскольку $n = 10\,000$, то из (32) получаем $\sigma_n \approx 0,004$, $3\sigma_n \approx 0,012$. Можно убедиться, что абсолютная величина погрешности действительно практически никогда не превышает 0,012. Следует заметить, что на практике, когда применяется метод Монте-Карло, точное значение вероятности p , конечно, не бывает известно. Поэтому вероятность p при расчёте $D(x)$ определяется приближённо, уже в ходе применения метода. В рассмотренном случае использование приближённой вероятности вместо точной вероятности не изменяет существенно выводов о погрешности расчёта.

Данный метод становится практически единственным методом, пригодным для численного расчёта интегралов. В то же время и при размерности $r = 1$ – в тех случаях, когда область интегрирования имеет сложную конфигурацию, – метод Монте-Карло может оказаться существенно более предпочтительным, чем пошаговые методы.

Сведение интеграла к математическому ожиданию.

Пусть требуется вычислить определенный интеграл вида:

$$I = \int_a^b h(t)dt, \quad (33)$$

с конечным или бесконечным интервалом интегрирования (a, b) . Чтобы применить метод Монте-Карло, перепишем интеграл (33) в форме (24) м. о. функции и рассчитаем это м. о., используя оценку (28).

Вначале введём под знак интеграла какую-либо п. р. в. $f(t)$:

$$I = \int_a^b h(t)dt = \int_a^b \frac{h(t)}{f(t)} f(t)dt. \quad (34)$$

Очевидно, выбранная функция $f(t)$ не должна быть равна нулю в точках интервала интегрирования (a, b) . Теперь для приведения интеграла к форме (24) остаётся, не меняя значения интеграла, перейти к бесконечным пределам интегрирования. Для этого определим вспомогательную функцию $H(t)$ через заданную подынтегральную функцию $h(t)$ так, чтобы $H(t)$ совпадала с $h(t)$ на интервале (a, b) , но была равна нулю вне этого интервала:

$$H(t) = \begin{cases} h(t) & \text{при } t \in (a, b), \\ 0 & \text{в противном случае.} \end{cases} \quad (35)$$

Тогда преобразование (34) можно продолжить так:

$$\int_a^b \frac{h(t)}{f(t)} f(t) dt = \int_a^b \frac{H(t)}{f(t)} f(t) dt = \int_{-\infty}^{\infty} \frac{H(t)}{f(t)} f(t) dt. \quad (36)$$

Представление (36) интеграла (33) является м. о. функции $[H(x)/f(x)]$ от с. в. $x \sim f(t)$ (сравните (36) с формулой (24)). Таким образом, искомый интеграл:

$$I = \int_a^b h(t) dt = M \left[\frac{H(x)}{f(x)} \right], \quad (37)$$

где $x \sim f(t)$,

т. е. может быть рассчитан методом Монте-Карло с использованием оценки (1.37). Для такого расчёта нужно только сгенерировать выборку с. в. x , имеющей п. р. в. $f(t)$, вычислить и усреднить функцию $[H(x)/f(x)]$ выборочных значений x .

Заметим, что если выбирается такая $f(t)$, которая равна нулю при всех $t \notin (a, b)$, то в представлении (37) вместо $H(x)$ можно использовать непосредственно функцию $h(x)$, поскольку в этом случае в генерируемой выборке x всегда попадает в (a, b) .

Управление скоростью сходимости.

От выбора функции $f(t)$ зависит скорость сходимости оценки интеграла к его точному значению. Скорость сходимости определяется главным образом дисперсией, усредняемой в (37) с. в. $y = H(x)/f(x)$. Чем меньше $D(y)$, тем меньше размах погрешностей оценки и тем меньшее число опытов n требуется для определения интеграла с требуемой точностью. В [4] даются рекомендации по

ускорению сходимости расчётов и приводится пример вычисления интеграла на полубесконечном интервале, где за счёт правильного выбора плотности $f(t)$ объём расчётов сокращается примерно в тысячу раз.

1.3 Постановка задачи разработки

Задачей данной ВКР является разработка программного обеспечения расчета кратных интегралов методом статистических испытаний применительно группы факторов воздействия к элементу СВ по неоднородной площади объекта.

1.4 Планирование разработки и оценка бюджета

Качественное и своевременное выполнение работ по любому проекту, в том числе и в сфере информационных технологий, невозможно при отсутствии грамотного планирования. Оно позволяет построить график выполнения работ над проектом таким образом, чтобы добиться минимизации времени и затрат на выполнение всех поставленных задач и целей с приемлемым качеством.

Планирование разработки и оценка бюджета была выполнена при помощи программы для управления проектами Microsoft Project. Проект был начат 28 марта 2023 года. При планировании разработки было выделено 16 задачи. Перечень задач и их описание представлено на таблице 2.

Таблица 2 – Перечень задач проекта

№	Название задачи	Длительность	Предшественники
1	Начало реализации проекта	0 дней	
2	Программирование	20,33 дней	
3	Разработка клиентской части приложения	17,33 дней	
4	Постановка задачи	4 дней	1
5	Разработка интерфейса	6 дней	4
6	Разработка клиентской части завершена	0 дней	4
7	Программирование завершено	0 дней	
8	Отладка	4,33 дней	
9	Тестирование и исправление ошибок	3 дней	
10	Составление программной документации	1 день	1
11	Отладка завершена	0 дней	10
12	Документирование	2 дней	
13	Создание установочного файла	1 день	"8;12"
14	Написание руководства пользования	0 дней	

Продолжение таблицы 3

15	Документация завершена	0 дней	13
16	Конец проекта	0 дней	21

В таблице 3 представлено описание затрачиваемых ресурсов.

Таблица 3 – Список ресурсов проекта

Название ресурса	Тип	Стандартная ставка	Ставка сверхурочных
Ведущий программист	Трудовой	250 руб/ч	500 руб/ч
Программист	Трудовой	250 руб/ч	400 руб/ч
Тестировщик	Трудовой	200 руб/ч	300 руб/ч
Интернет	Затраты	600 руб	-
Лицензия программы	Затраты	1500 руб	-

Распределение ресурсов по задачам представлено в таблице 4.

Таблица 4 – Назначение ресурсов

№	Название задачи	Ресурс	Единицы
1	Начало реализации проекта	-	-
2	Программирование	-	-
3	Разработка клиентской части приложения	Программист Интернет	30% 600 руб
4	Постановка задачи	Ведущий программист	70%
5	Разработка интерфейса	Ведущий программист Программист	100% 100%
6	Разработка клиентской части завершена	Ведущий программист Программист	100% 100%
7	Программирование завершено	-	-
8	Отладка	Тестировщик Программист	70% 30%
9	Тестирование и исправление ошибок	-	-
10	Составление программной документации	Ведущий программист Программист	50% 50%
11	Отладка завершена	-	-
12	Документирование	Ведущий программист Тестировщик	50% 50%
13	Создание установочного файла	Программист	50%
14	Написание руководства пользования	Программист	50%
15	Документация завершена	-	-
16	Конец проекта	-	-

Таким образом, реализацией проекта будут заниматься ведущий программист, программист и тестировщик, которым для работы нужна лицензия и доступ в интернет. Для вычисления длительности реализации проекта и его критического пути построен сетевой график (рисунок 7).

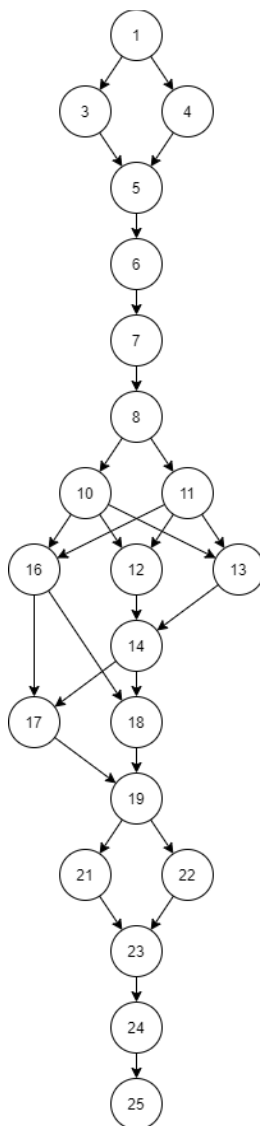


Рисунок 7 – Сетевой график проекта

Перечень задач в Microsoft Project представлен на рисунке 8.

	Название задачи	Длительность	Начало	Окончание	Предшественники
1	Начало реализации проекта	0 дней	Пн 05.12.22	Пн 05.12.22	
2	Программирование	20,33 дней	Вт 06.12.22	Вт 03.01.23	
3	Разработка клиентской части приложения	17,33 дней	Вт 06.12.22	Чт 29.12.22	
4	Постановка задачи	4 дней	Вт 06.12.22	Пт 09.12.22	1
5	Разработка интерфейса	6 дней	Пн 12.12.22	Пн 19.12.22	4
6	Разработка клиентской части завершена	0 дней	Пн 19.12.22	Пн 19.12.22	4
7	Программирование завершено	0 дней	Чт 22.12.22	Чт 22.12.22	
8	Отладка	4,33 дней	Пт 23.12.22	Чт 29.12.22	
9	Тестирование и исправление ошибок	3 дней	Ср 28.12.22	Пн 02.01.23	
10	Составление программной документации	1 день	Ср 28.12.22	Чт 29.12.22	1
11	Отладка завершена	0 дней	Чт 29.12.22	Чт 29.12.22	10
12	Документирование	2 дней	Чт 29.12.22	Пн 02.01.23	
13	Создание установочного файла	1 день	Пн 02.01.23	Вт 03.01.23	8;12
14	Написание руководства пользователя	0 дней	Вт 03.01.23	Вт 03.01.23	
15	Документация завершена	0 дней	Вт 03.01.23	Вт 03.01.23	13
16	Конец проекта	0 дней	Вт 03.01.23	Вт 03.01.23	15
17	Отладка завершена	0 дней	Вт 03.01.23	Вт 03.01.23	16
18	Документирование	2 дней	Вт 03.01.23	Чт 05.01.23	
19	Создание установочного файла	1 день	Вт 03.01.23	Ср 04.01.23	17
20	Написание руководства пользователя	1 день	Ср 04.01.23	Чт 05.01.23	19
21	Документация завершена	0 дней	Чт 05.01.23	Чт 05.01.23	20
22	Конец проекта	0 дней	Чт 05.01.23	Чт 05.01.23	21

Рисунок 8 – Перечень задач проекта

Диаграмма Ганта, спроектированная в Microsoft Project представлена на рисунке 9.

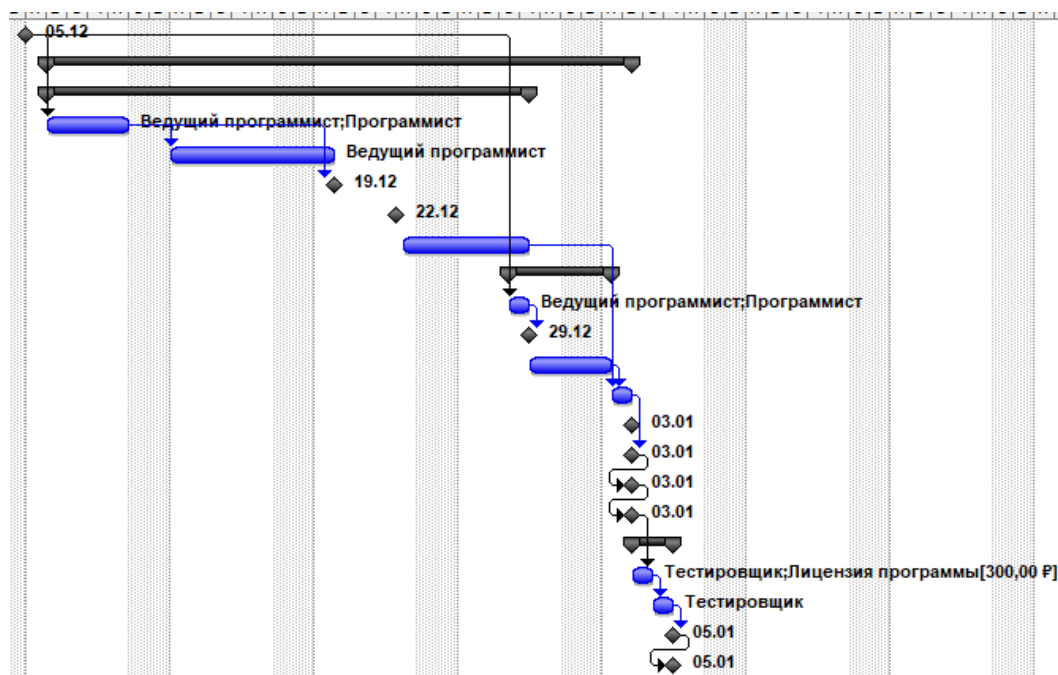


Рисунок 9 – Диаграмма Ганта

На рисунке 10 представлен обзорный отчет проекта. Реализация проекта займет 67 дня, а предполагаемые затраты 16250 рублей.

<u>Даты</u>			
Начало:	Пн 12.12.22	Окончание:	Вт 14.03.23
Баз. начало:	НД	Баз. окончание:	НД
Факт. начало:	НД	Факт. окончание:	НД
Отклон. начала:	0 дней	Отклон. окончания:	0 дней
<u>Длительность</u>			
Текущий план:	67 дней?	Оставш аяся:	67 дней?
Базовый план:	0 дней?	Фактическая:	0 дней
Отклонение:	67 дней?	Процент завершения	0%
<u>Трудозатраты</u>			
Текущий план:	64 ч	Оставш иеся:	64 ч
Базовый план:	0 ч	Фактические:	0 ч
Отклонение:	64 ч	Процент завершения	0%
<u>Затраты</u>			
Текущий план:	16 250,00 Р	Оставш иеся:	16 250,00 Р
Базовый план:	0,00 Р	Фактические:	0,00 Р
Отклонение:	16 250,00 Р		
<u>Состояние задачи</u>		<u>Состояние ресурса</u>	
Еще не начатые задачи:	23	Трудовые ресурсы:	5
Выполняющиеся задачи:	0	Трудовые ресурсы с прев. дост	0
Заверш енные задачи:	0	Материальные ресурсы:	0
Итого задач:	23	Итого ресурсов:	5

Рисунок 10 – Обзорный отчет проекта

На основе произведенных оценок стоимости и сроков проекта можно сделать вывод о его экономической целесообразности.

2 Сущность метода Монте-Карло и моделирование случайных величин

Предположим, что нам необходимо вычислить площадь плоской фигуры S . Это может быть произвольная фигура, заданная графически или аналитически (связная или состоящая из нескольких частей). Пусть это будет фигура, заданная на рисунке 11.

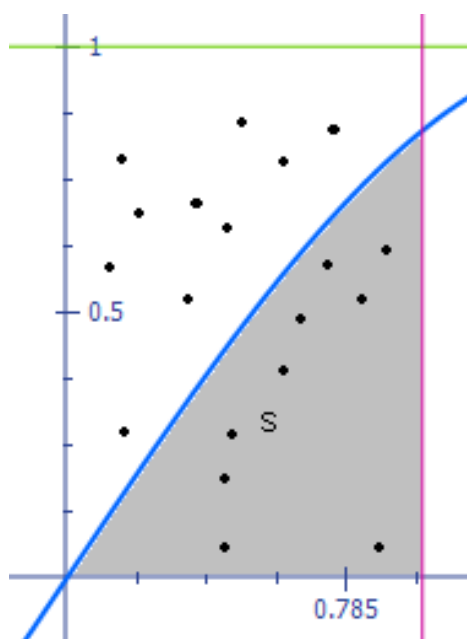


Рисунок 11 – Определение площади плоской фигуры

Предположим, что эта фигура расположена внутри единичного квадрата.

Выберем внутри квадрата N случайных точек. Обозначим через N_1 число точек, попавших внутрь фигуры S . Геометрически видно, что площадь фигуры S приближенно равна отношению N_1/N . Причем, чем больше число N , тем больше точность этой оценки.

Для того чтобы выбирать точки случайно, необходимо перейти к понятию случайная величина. Случайная величина ε непрерывная, если она может принимать любое значение из некоторого интервала (a,b) .

Непрерывная случайная величина ε определяется заданием интервала (a,b) , содержащего возможные значения этой величины, и функции $p(x)$,

которая называется плотностью вероятностей случайной величины ε (плотностью распределения ε). Физический смысл $p(x)$ следующий: пусть (a', b') - произвольный интервал, такой что $a \leq a' < b' \leq b$, тогда вероятность того, что ε окажется в интервале (a', b') , равна интегралу:

$$P\{a' < \varepsilon < b'\} = \int_{a'}^{b'} p(x) dx. \quad a \leq a' < b' \leq b \quad (38)$$

Множество значений ε может быть любым интервалом (возможен случай $a = -\infty, b = \infty$). Однако плотность $p(x)$ должна удовлетворять двум условиям:

- плотность $p(x)$ положительна:

$$p(x) > 0;$$

- интеграл от плотности $p(x)$ по всему интервалу (a, b) равен 1:

$$\int_a^b p(x) dx = 1.$$

Математическим ожиданием непрерывной случайной величины называется число:

$$M\varepsilon = \int_a^b xp(x) dx.$$

Дисперсией непрерывной случайной величины называется число:

$$D = \int_a^b (x - M\varepsilon)^2 p(x) dx$$

Нормальной случайной величиной называется случайная величина ζ , определённая на всей оси $(-\infty, \infty)$ и имеющая плотность:

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left\{-\frac{(x-a)^2}{2\sigma^2}\right\},$$

где a, σ - числовые параметры.

Любые вероятности вида $P\{x' < \zeta < x''\}$ легко вычисляются с помощью таблицы, в которой приведены значения функции:

$$\Phi(x) = \frac{2}{\sqrt{2\pi}} \int_0^x e^{-t^2/2} dt ,$$

называемой обычно интегралом вероятностей.

Согласно (38):

$$P\{x' < \zeta < x''\} = \frac{1}{\sigma\sqrt{2\pi}} \int_{x'}^{x''} \exp\left\{-\frac{(x-a)^2}{2\sigma^2}\right\} dx .$$

В интеграле сделаем замену переменной $x-a = \sigma t$, тогда получим, что:

$$P\{x' < \zeta < x''\} = \frac{1}{\sqrt{2\pi}} \int_{t_1}^{t_2} e^{-t^2/2} dt ,$$

где $t_1 = (x' - a)/\sigma, t_2 = (x'' - a)/\sigma$.

Отсюда следует, что $P\{x' < \zeta < x''\} = 0.5[\Phi(t_2) - \Phi(t_1)]$. Также $\Phi(-x) = -\Phi(x)$.

Нормальные случайные величины очень часто встречаются при исследовании самых различных по своей природе вопросов.

Выбрав $x' = a - 3\sigma$, $x'' = a + 3\sigma$, найдём $t_1 = -3, t_2 = 3$. Следовательно,

$$P\{a - 3\sigma < \zeta < a + 3\sigma\} = \Phi(3) = 0.997 .$$

Вероятность 0.997 настолько близка к 1, что иногда последнюю формулу интерпретируют так: при одном испытании практически невозможно получить значение ζ , отличающееся от $M\zeta$ больше чем на 3σ .

Проводя большое количество опытов, и получая большое количество случайных величин можно воспользоваться центральной предельной теоремой теории вероятностей. Эта теорема впервые была сформулирована П. Лапласом. Обобщением этой теоремы занимались многие выдающиеся математики, в том числе П.Л. Чебышёв, А.А. Марков, А.М. Ляпунов. Её доказательство достаточно сложно [4].

Рассмотрим N одинаковых независимых случайных величин $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N$, так что распределения вероятностей этих величин совпадают. Следовательно, их математические ожидания и дисперсии также совпадают. Величины эти могут быть как непрерывными, так и дискретными.

Обозначим, что:

$$\begin{aligned} M\varepsilon_1 = M\varepsilon_2 = \dots = M\varepsilon_N = m \\ D\varepsilon_1 = D\varepsilon_2 = \dots = D\varepsilon_N = b^2 \end{aligned} .$$

Сумму всех этих величин обозначим через $\rho_N = \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N$.

Используя соотношения

$$\begin{aligned} M(\varepsilon + \eta) = M\varepsilon + M\eta \\ D(\varepsilon + \eta) = D\varepsilon + D\eta \end{aligned} ,$$

получаем:

$$\begin{aligned} M\rho_N = M(\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N) = Nm, \\ D\rho_N = D(\varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N) = Nb^2 \end{aligned} .$$

Рассмотрим теперь нормальную случайную величину ζ_N с такими же параметрами: $a = Nm, \sigma = b\sqrt{N}$.

В центральной предельной теореме утверждается, что для любого интервала (a', b') при больших N :

$$P\{a' < \rho_N < b'\} \approx \int_{a'}^{b'} \rho_{\zeta_N}(x) dx .$$

Смысл этой теоремы в том, что сумма ρ_N большого числа одинаковых случайных величин приближенно нормальна. На самом деле эта теорема справедлива при гораздо более широких условиях: все слагаемые не обязаны быть одинаковыми и независимыми; существенно только, чтобы отдельные слагаемые не играли большой роли в сумме. Эта теорема оправдывает часто встречающиеся нормальные случайные величины. В самом деле, когда встречается суммарное воздействие большого числа незначительных случайных факторов, результирующая случайная величина оказывается нормальной.

2.1 Общий алгоритм расчета кратных интегралов методом Монте-Карло

Используя эти данные из теории вероятностей можно перейти к описанию общей схемы метода Монте-Карло. Допустим, что требуется вычислить какую-то неизвестную величину m . Попробуем придумать такую случайную величину ε , чтобы $M\varepsilon = m$. Пусть при этом $D\varepsilon = b^2$.

Рассмотрим N независимых случайных величин $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N$ распределения которых совпадают с распределением ε . Если N достаточно велико, то, согласно центральной предельной теореме, распределение суммы $\rho_N = \varepsilon_1 + \varepsilon_2 + \dots + \varepsilon_N$ будет приблизительно нормальным с параметрами $a = Nm, \sigma = b\sqrt{N}$. Из (54) следует, что $P\{Nm - 3b\sqrt{N} < \rho_N < Nm + 3b\sqrt{N}\} \approx 0.997$.

Последнее соотношение перепишем в виде:

$$P\left\{\left|\frac{1}{N} \sum_{j=1}^N \varepsilon_j - m\right| < \frac{3b}{\sqrt{N}}\right\} \approx 0.997 \quad (39)$$

Это соотношение даёт и метод расчёта m , и оценку погрешности.

В самом деле, найдём N значений случайной величины ε . Из (39) видно, что среднеарифметическое этих значений будет приближенно равно m . С большой вероятностью погрешность приближения не превосходит величины $3b/\sqrt{N}$. Эта погрешность стремится к нулю с ростом N . На практике часто используют не оценку сверху $3b/\sqrt{N}$, а ее вероятную ошибку, которая приближенно равна $r_N = 0.6745b/\sqrt{N}$. Именно такой обычно порядок фактической погрешности расчёта, которая равна:

$$\left|\frac{1}{N} \sum_{j=1}^N \varepsilon_j - m\right|.$$

Для получения случайных чисел используют обычно три способа: таблицы случайных величин, генераторы случайных чисел и метод псевдослучайных чисел.

Таблицы случайных чисел используют предпочтительно при расчётах вручную. Определяющую роль в этом играют два факта:

- при использовании ЭВМ легче и удобней воспользоваться генератором случайных чисел, получаемых тут же, чем загружать из памяти значения таблицы, которая к тому же, будет занимать там место;
- при подсчёте вручную нет необходимости использовать ЭВМ, так как часто необходимо выяснить лишь порядок искомой величины [5].

Генераторы случайных чисел анализируют какой-либо процесс, доступный для них (шумы в электронных лампах, скачки напряжения) и составляют последовательность из 0 и 1, из которых составляются числа с определёнными разрядами, однако такой метод получения случайных величин имеет свои недостатки. Во-первых, трудно проверить вырабатываемые числа. Проверки приходится делать периодически, так как из-за каких-либо неисправностей может возникнуть так называемый дрейф распределения (нули и единицы в каком-либо из разрядов станут появляться не одинаково часто). Во-вторых, обычно все расчёты на ЭВМ проводятся несколько раз, чтобы исключить возможность сбоя. Но воспроизвести те же самые случайные числа невозможно, если их только не запоминать по ходу счёта. А если запоминать, то снова появляется случай таблиц.

Таким образом, самым эффективным способом получения случайных чисел – это использование псевдослучайных чисел.

Числа, получаемые по какой-либо формуле и имитирующие значения случайной величины γ , называются псевдослучайными числами.

Первый алгоритм для получения псевдослучайных чисел был предложен Дж. Нейманом. Он называется методом середины квадратов.

Пусть задано четырехзначное число $\gamma_0 = 0.8754$. Возведём его в квадрат. Получим восьмизначное число $\gamma_0^2 = 0.76632516$. Выберем четыре средние цифры этого числа и положим $\gamma_1 = 0.6325$. Далее $\gamma_1^2 = 0.40005625$ и т. д.

Но этот алгоритм не оправдал себя, так как получается слишком много малых значений. Поэтому были разработаны другие алгоритмы. Наибольшее

распространение получил алгоритм, называемый методом сравнений (Д. Лемер): определяется последовательность целых чисел m_k , в которой начальное число $m_0 = 1$ задано, а все последующие числа $m_1, m_2, \dots$ вычисляются по одной и той же формуле:

$$m_{k+1} = 5^{17} m_k \pmod{2^{40}} \text{ при } k = 0, 1, 2, \dots \quad (40)$$

По числам m_k вычисляются псевдослучайные числа:

$$\gamma_k = 2^{-40} m_k. \quad (41)$$

Формула (40) означает, что число m_{k+1} равно остатку, полученному при делении $5^{17} m_k$ на 2^{40} , такой остаток называют наименьшим положительным вычетом по модулю 2^{40} . Формулы (40), (41) легко реализовать на ЭВМ.

Достоинства метода псевдослучайных чисел довольно очевидны. Во-первых, на получение каждого числа затрачивается всего несколько простых операций, так что скорость генерирования случайных чисел имеет тот же порядок, что и скорость работы ЭВМ. Во-вторых, программа занимает не так много места в памяти. В-третьих, любое из чисел γ_k может быть легко воспроизведено. В-четвёртых, необходимо лишь один раз проверить «качество» такой последовательности, затем её можно много раз безбоязненно использовать при расчёте однотипных задач.

Единственный недостаток метода — ограниченность количества псевдослучайных чисел, так как если последовательность чисел $\gamma_0, \gamma_1, \dots, \gamma_k, \dots$ вычисляется на ЭВМ по формуле вида:

$$\gamma_{k+1} = F(\gamma_k),$$

то эта последовательность обязательно периодическая. Впрочем, для наиболее распространённых псевдослучайных чисел период столь велик, что превосходит любые практические потребности. Подавляющее большинство расчётов по методу Монте-Карло осуществляется с использованием псевдослучайных чисел.

Значения любой случайной величины можно получить путём преобразования значений одной какой-либо случайной величины. Обычно роль такой случайной величины играет случайная величина γ , равномерно распределённая в интервале $(0,1)$. Процесс нахождения значения какой-либо случайной величины ε путём преобразования одного или нескольких значений γ называется разыгрыванием случайной величины ε .

Допустим, что необходимо получать значения случайной величины ε , распределённой в интервале (a,b) , с плотностью $p(x)$. Докажем, что значения ε можно находить из уравнения:

$$\int_a^{\varepsilon} p(x)dx = \gamma, \quad (42)$$

т. е. выбрав очередное значение γ , надо решить уравнение (42) и найти очередное значение ε .

Для доказательства рассмотрим функцию:

$$y = \int_a^x p(x)dx.$$

Из общих свойств плотности (1.2), (1.3) следует, что:

$$y(a) = 0, y(b) = 1, y'(x) = p(x) > 0.$$

Значит, функция $y(x)$ монотонно возрастает от 0 до 1, и любая прямая $y = \gamma$, где $0 < \gamma < 1$, пересекает график $y = y(x)$ в одной единственной точке, абсциссу которой мы и принимаем за ε . Таким образом, уравнение (42) всегда имеет одно и только одно решение.

Выберем теперь произвольный интервал (a',b') , содержащийся внутри (a,b) . Точкам этого интервала $a' < x < b'$ отвечают ординаты кривой $y = y(x)$, удовлетворяющие неравенству $y(a') < y < y(b')$.

Поэтому, если ε принадлежит интервалу $a' < x < b'$, то γ принадлежит интервалу $y(a') < y < y(b')$, и наоборот. Значит $P\{a' < \varepsilon < b'\} = P\{y(a') < \gamma < y(b')\}$.

Так как γ равномерно распределена в $(0,1)$, то:

$$P\{y(a') < \gamma < y(b')\} = y(b') - y(a') = \int_{a'}^{b'} p(x) dx,$$

$$P\{a' < \varepsilon < b'\} = \int_{a'}^{b'} p(x) dx,$$

а это и означает, что случайная величина ε , являющаяся корнем уравнения (42), имеет плотность вероятностей $p(x)$.

Может оказаться, что разрешить уравнение (42) относительно ε трудно, например, в случаях, когда интеграл от $p(x)$ не выражается через элементарные функции или когда плотность $p(x)$ задана графически. Предположим, что случайная величина ε определена на конечном интервале (a, b) и плотность её ограничена $p(x) \leq M_0$.

Разыгрывать значение ε можно следующим образом:

- выбираются два значения γ' и γ'' случайной величины γ и строится случайная точка $\Gamma(\eta', \eta'')$ с координатами:

$$\eta' = a + \gamma'(b - a), \eta'' = \gamma'' M_0;$$

- если точка Γ лежит под кривой $y = p(x)$, то полагаем $\varepsilon = \eta'$, если же точка Γ лежит над кривой $y = p(x)$, то пара (γ', γ'') отбрасывается и выбирается новое значение.

2.2 Вычисление интегралов

Рассмотрим функцию $g(x)$, заданную на интервале $a < x < b$, требуется приближенно вычислить интеграл:

$$I = \int_a^b g(x) dx. \quad (43)$$

Этот интеграл может быть несобственным, но абсолютно сходящимся.

Выберем произвольную плотность распределения $p(x)$, определённую на интервале (a, b) . Наряду со случайной величиной ε , определённой в интервале (a, b) с плотностью $p(x)$, необходимо определить случайную величину:

$$\eta = \frac{g(\varepsilon)}{p(\varepsilon)}.$$

Согласно соотношению $Mf(\varepsilon) = \int_a^b f(x)p(x)dx$ получим:

$$M\eta = \int_a^b \frac{g(x)}{p(x)} p(x)dx = I.$$

Рассмотрим теперь N одинаковых независимых случайных величин $\eta_1, \eta_2, \dots, \eta_N$ и применим к их сумме центральную предельную теорему. Формула (43) в этом случае запишется так:

$$P \left\{ \left| \frac{1}{N} \sum_{j=1}^N \eta_j - I \right| < 3\sqrt{\frac{D_\eta}{N}} \right\} \approx 0.997.$$

Последнее соотношение означает, что если выбирать N значений $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_N$, то при достаточно большом N :

$$\frac{1}{N} \sum_{j=1}^N \frac{g(\varepsilon_j)}{p(\varepsilon_j)} \approx I. \quad (44)$$

Оно показывает также, что с очень большой вероятностью погрешность приближения (44) не превосходит $3\sqrt{\frac{D_\eta}{N}}$.

Для расчёта интеграла (43) можно использовать любую случайную величину ε . Определённую в интервале (a, b) с плотностью $p(x) > 0$. В любом случае $M\eta = M[g(\varepsilon)/p(\varepsilon)] = I$. Однако дисперсия $D\eta$, а с ней и оценка погрешности формулы (44) зависят от того, какая величина ε используется, так как:

$$D\eta = M\eta^2 - I^2 = \int_a^b \frac{g^2(x)}{p(x)} dx - I^2. \quad (45)$$

Докажем, что это выражение будет минимальным тогда, когда $p(x)$ пропорциональна $|g(x)|$.

Для этого воспользуемся неравенством:

$$\left(\int_a^b |u(x)v(x)| dx \right)^2 \leq \int_a^b u^2(x) dx \int_a^b v^2(x) dx, \text{ в котором положим } u = g(x)/\sqrt{p(x)},$$

$v = \sqrt{p(x)}$. Получим неравенство:

$$\left(\int_a^b |g(x)| dx \right)^2 \leq \int_a^b \frac{g^2(x)}{p(x)} dx \int_a^b p(x) dx = \int_a^b \frac{g^2(x)}{p(x)} dx. \quad (46)$$

Из (45) и (46) следует, что:

$$D_\eta \geq \left(\int_a^b |g(x)| dx \right)^2 - I^2. \quad (47)$$

Остается доказать, что нижняя граница дисперсии (47) реализуется при выборе плотности $p_0(x) = c|g(x)|$. Так как:

$$c = \left(\int_a^b |g(x)| dx \right)^{-1}.$$

Следовательно,

$$\int_a^b \frac{g^2(x)}{p_0(x)} dx = \frac{1}{c} \int_a^b |g(x)| dx = \left(\int_a^b |g(x)| dx \right)^2,$$

и правая часть (45) обращается в правую часть (47).

Использовать плотность $p_0(x)$ для расчёта практически невозможно, так как для этого нужно знать значение интеграла $\int_a^b |g(x)| dx$. А его вычисление представляет собой задачу, равноценную задаче о вычислении интеграла (47). Поэтому ограничиваются следующей рекомендацией: желательно, чтобы плотность $p(x)$ была пропорциональна $|g(x)|$.

Конечно, выбирать очень сложные $p(x)$ нельзя, так как процедуры разыгрывания ε станет очень трудоёмкой. Оценку (44) с плотностью $p(x)$, сходной $|g(x)|$, называют существенной выборкой.

Также, если стоит задача вычислить интеграл (43), преобразуем его к виду:

$$\int_a^b \frac{g(x)}{p(x)} p(x) dx = I \quad (48)$$

Если теперь обозначить:

$$\frac{g(x)}{p(x)} = h^*(x),$$

то интеграл принимает вид:

$$I = \int_a^b h^*(x) p(x) dx,$$

и может быть вычислен при помощи метода статистических испытаний.

В частном случае, если a и b конечны или их можно считать конечными приближенно, в качестве $f(x)$ целесообразно выбрать равномерный закон распределения.

Как известно, плотность вероятности равномерного закона распределения в интервале (a, b) равна:

$$p(x) = \frac{1}{b-a} \quad (49)$$

Подставим в интеграл (48) значение $p(x)$ из формулы (49) и получим:

$$I = (b-a) \int_a^b g(x) \frac{1}{b-a} dx \quad (50)$$

и рассмотрим процедуру вычисления: из множества равномерно распределённых случайных чисел выбирается x_i . Для каждого значения x_i вычисляется $g(x_i)$, затем вычисляется среднее значение,

$$\bar{g}(x_i) = \frac{1}{N} \sum_{i=1}^N g(x_i),$$

функции $g(x)$ на интервале (a, b) .

Таким образом, величина интеграла (50) может быть представлена в виде следующей формулы:

$$I \approx (b-a) \bar{g}(x_i).$$

Пусть из общего числа N случайных точек n точек принадлежат объёму v , тогда имеет место приближенная формула:

$$I \approx B \frac{n}{N}.$$

3 Программное обеспечение расчета кратных интегралов методом статистических испытаний при расчете вероятности воздействия на противодействующую систему

В настоящее время для автоматизации математических расчетов применяются различные системы программ и пакетов программ, такие как Mathcad, Mathematica, MapleV и др. Среди них особое место занимает интегрированная система компьютерной математики MATLAB. Она широко используется как в учебном процессе, так и в научно-исследовательских и опытно-конструкторских разработках во многих вузах и на предприятиях различных отраслей промышленности. Первые версии системы были разработаны в начале 1980-х гг. в компании MathWorks для компьютеров класса IBMPC, VAX и Macintosh. В настоящее время система MATLAB может работать с множеством популярных компьютерных платформ. Система MATLAB специально создана для проведения именно инженерных расчетов и опирается на матричные операции при решении задач линейной алгебры и математическом моделировании статических и динамических систем и объектов.

MATLAB – это высокоуровневый язык и интерактивная среда программирования, численных расчетов и визуализации результатов. С помощью MATLAB можно анализировать данные, разрабатывать алгоритмы, создавать модели и приложения [8]. Основу системы составляет высокоуровневый интерпретируемый язык программирования MATLAB, ориентированный на математические расчеты, который как по скорости вычислений, так и по способности адаптации системы к решению самых разнообразных задач пользователей во многом превосходит традиционные языки программирования. Язык программирования MATLAB содержит несколько сотен функций для анализа данных в различных областях математики, а также широкий набор функции линейной алгебры, быстрого преобразования Фурье, функции для работы с полиномами, функции базовой статистики и численного решения дифференциальных уравнений и т. д. В

составе пакета MATLAB имеется большое количество функций для построения графиков, в том числе трехмерных, визуального анализа данных и создания анимированных роликов. Встроенная среда разработки позволяет создавать графические интерфейсы пользователя с различными элементами управления, такими как кнопки, поля ввода и т. п.

MATLAB предоставляет удобные средства для разработки алгоритмов, включая высокоуровневые с использованием концепций объектно-ориентированного программирования. В MATLAB имеются все необходимые средства интегрированной среды разработки, включая отладчик и профайлер. MATLAB позволяет создавать специальные наборы инструментов (англ. *toolbox*), которые расширяют его функциональные возможности. Они представляют собой коллекции функций (*M*-файлов), написанных на языке MATLAB для решения определенного класса задач. Компания MathWorks поставляет наборы инструментов, которые используются в таких областях, как автомобилестроение, связь, электроника и полупроводники, энергетика, автоматизация и машиностроение, авиакосмос и ВПК, биотехнологии и фармакология, финансы и т. д. [9]. MATLAB является открытой и расширяемой системой. Большинство команд и функций системы реализованы в виде *M*-файлов текстового формата (с расширением «.m») и файлов на языке Си, причем все файлы доступны для модификации. С системой MATLAB могут интегрироваться такие популярные математические системы, как Mathematica, Mathcad и MapleV. MATLAB позволяет работать как в режиме мощного научного калькулятора, так и составлять собственные программы для проведения научных исследований. Он легко интегрируется с текстовым редактором Word, что позволяет включать сложные расчеты и графики в текст инженерных и научных отчетов и статей. В настоящее время широко используется версия MATLAB R2022b, интерфейс которой содержит окно редактирования и интерфейсную ленту, на вкладках которой сгруппированы связанные команды. С целью преемственности с более ранними версиями MATLAB в MATLAB R2022b после указания пути доступа к

$$\vec{\Delta}_{ijllkrst} = \vec{\gamma}_{ij} + \vec{\Delta}_{\varepsilon\partial} + \vec{\Delta}_{\partial_i} + \vec{\Delta}_{\partial_{st}} + \vec{\Delta}_{BM_{rst}} + \vec{\Delta}_{pc_{krst}} + \vec{\Delta}_{\partial_{llkrst}}, \quad (52)$$

где $\vec{\gamma}_{ij}$ – неслучайный вектор, определяющий положение i,j -й точки цели C_{ij} ;

$\vec{\Delta}_{\varepsilon\partial}$ – повторяющийся случайный вектор, связанный с меткостью воздействия и отражающий смещение центра группирования точек вскрытия элементов СВ в точку C и повторяющийся для всех элементов СВ, выпущенных группой фактора воздействия;

$\vec{\Delta}_{\partial_i}$ – повторяющийся случайный вектор, связанный с меткостью воздействия и отражающий смещение центра группирования точек элементов СВ t -го фактора воздействия в точку C_t и повторяющийся для всех элементов СВ, выпущенных t -м фактором воздействия;

$\vec{\Delta}_{\partial_{st}}$ – повторяющийся случайный вектор, связанный с меткостью воздействия и отражающий смещение центра группирования точек элементов СВ s -й группы t -го фактора воздействия в точку C_{st} и повторяющийся для всей s -ой группы СВ t -го фактора воздействия;

$\vec{\Delta}_{BM_{rst}}$ – повторяющийся случайный вектор, связанный с меткостью воздействия и отражающий смещение центра группирования точек элементов r -й средства воздействия s -ой группы СВ t -го фактора воздействия в точку C_{rst} и повторяющийся для всех элементов СВ, выпущенных r -й элемента s -й группы СВ t -го фактор воздействия;

$\vec{\Delta}_{pc_{krst}}$ – неповторяющийся случайный вектор, связанный с кучностью воздействия и характеризующий смещение точки вскрытия СВ k -го элемента от центра группирования C_{rst} в точку R_{krst} при организации процедур воздействия из r -й элемента s -й группы СВ t -го фактора воздействия;

$\vec{\Delta}_{\partial_{llkrst}}$ – неповторяющийся случайный вектор, связанный с кучностью воздействия и характеризующий смещение точки разрыва l -го подэлемента СВ от точки вскрытия k -го элемента СВ R_{krst} в точку R_{llkrst} при организации

процедур воздействия из r -й элемента s -й 1-я группа СВ t -го фактор воздействия [5].

Векторы $\vec{\Delta}_{\varepsilon\partial}, \vec{\Delta}_{\partial_t}, \vec{\Delta}_{\delta_{st}}, \vec{\Delta}_{BM_{rst}}, \vec{\Delta}_{pc_{krst}}, \vec{\Delta}_{\delta\partial_{lkrst}}$ являются независимыми случайными векторами, составляющие которых распределены по нормальным законам. По этой причине воздействия формирования элемент СВ отвечает схема шести групп ошибок.

Проектируя векторное равенство (52) на оси системы OXZ, получим:

$$\begin{cases} X_{ijklrst} = \gamma_{X_{ij}} + X_{\varepsilon\partial} + X_{\partial_t} + X_{\delta_{st}} + X_{BM_{rst}} + X_{pc_{krst}} + X_{\delta\partial_{lkrst}}, \\ Z_{ijklrst} = \gamma_{Z_{ij}} + Z_{\varepsilon\partial} + Z_{\partial_t} + Z_{\delta_{st}} + Z_{BM_{rst}} + Z_{pc_{krst}} + Z_{\delta\partial_{lkrst}}, \end{cases} \quad (53)$$

где $\gamma_{X_{ij}}, \gamma_{Z_{ij}}$ – координаты i, j -й точки объекта;

$X_{гд}, Z_{гд}$ – случайные величины, характеризующие повторяющиеся ошибки группы фактор воздействия, сохраняющие свои значения при всех организациях групп факторов воздействия;

$X_{дт}, Z_{дт}$ – случайные величины, характеризующие фактор воздействия повторяющихся ошибок, сохраняющие свои значения при всех организациях процедур воздействия t -го фактора воздействия;

$X_{\delta_{st}}, Z_{\delta_{st}}$ – случайные величины, характеризующие 1-ую группу СВ повторяющиеся ошибки, не изменяющие своих значений при всех организациях процедур воздействия s -ой группы СВ t -го фактора воздействия;

$X_{BM_{rst}}, Z_{BM_{rst}}$ – случайные величины, характеризующие повторяющиеся ошибки средство воздействия, не изменяющие своих значений при всех организациях процедур воздействия r -го элемента s -й группы СВ t -го фактора воздействия;

$X_{pc_{krst}}, Z_{pc_{krst}}$ – случайные величины, характеризующие отклонение точки вскрытия ВС k -го элемента СВ от центра группирования при организациях процедур воздействия из r -го элемента s -й группы СВ t -го фактора воздействия;

$X_{\bar{\sigma}_{lkrst}}, Z_{\bar{\sigma}_{lkrst}}$ – случайные величины, характеризующие отклонение точки разрыва 1-го подэлемента СВ от точки вскрытия k-го элемента СВ при организации процедур воздействия из r-го элемента s-й группы СВ t-го фактора воздействия.

Принимая допущение о нормальности случайных величин, входящих в (53) и переходя к срединным отклонениям, получим:

$$\begin{cases} E_x^2 = E_{x_{z\partial}}^2 + E_{x_{\partial t}}^2 + E_{x_{\bar{\sigma}_{st}}}^2 + E_{x_{BMrst}}^2 + B\delta_{pcrst}^2 + B\delta_{\bar{\sigma}_{lkrst}}^2, \\ E_z^2 = E_{z_{z\partial}}^2 + E_{z_{\partial t}}^2 + E_{z_{\bar{\sigma}_{st}}}^2 + E_{z_{BMrst}}^2 + B\delta_{pcrst}^2 + B\delta_{\bar{\sigma}_{lkrst}}^2, \end{cases}$$

где $E_{x_{rd}}^2, E_{z_{rd}}^2$ – дисперсии ошибок подготовки группы факторов воздействия;

$E_{x_{lt}}^2, E_{z_{lt}}^2$ – дисперсии ошибок подготовки t-го факторов воздействия, входящего в группу факторов воздействия;

$E_{x_{\bar{\sigma}_{st}}}^2, E_{z_{\bar{\sigma}_{st}}}^2$ – дисперсии ошибок подготовки s-й группы СВ t-го фактора воздействия;

$E_{x_{BMrst}}^2, E_{z_{BMrst}}^2$ – дисперсии ошибок подготовки r-го элемента s-й группы СВ t-го фактора воздействия;

$B\delta_{pcrst}^2, B\delta_{\bar{\sigma}_{lkrst}}^2$ – дисперсии ошибок рассеивания точки вскрытия СВ k-го элемента от центра группирования при организации процедур воздействия из r-го средства воздействия s-й группы СВ t-го фактора воздействия;

$B\delta_{\bar{\sigma}_{lkrst}}^2, B\delta_{\bar{\sigma}_{lkrst}}^2$ – дисперсии ошибок рассеивания точки распада 1-го подэлемента СВ от точки вскрытия k-го элемента СВ при организации процедур воздействия из r-го элемента s-й группы СВ t-го фактора воздействия.

Для любых показателей эффективности воздействие, определяемых в схеме нескольких групп ошибок, справедлива формула:

$$G = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(\bar{x}_n, \bar{z}_n) \varphi(\bar{x}_n, \bar{z}_n) d\bar{x}_n d\bar{z}_n, \quad (54)$$

где $G(\bar{x}_n, \bar{z}_n)$ - условная величина рассматриваемого показателя эффективности воздействия, отвечающая фиксированным величинам $\bar{x}_n(x_{гд}, x_{дт}, x_{бст}, x_{БМ_{rst}})$ и $\bar{z}_n(z_{гд}, z_{дт}, z_{бст}, z_{БМ_{rst}})$ повторяющихся ошибок;

$\varphi(\bar{x}_n, \bar{z}_n)$ – функция плотности вероятности повторяющихся ошибок.

Для проведения расчетов в схеме шести групп ошибок необходимо задать величины срединных отклонений:

$$E_{x_{гд}}, E_{z_{гд}}, E_{x_{дт}}, E_{z_{дт}}, E_{x_{бст}}, E_{z_{бст}}, E_{x_{БМ_{rst}}}, E_{z_{БМ_{rst}}}, B_{дгкрст}, B_{бгкрст}, B_{дбкрст}, B_{ббкрст}.$$

Для вычисления показателя эффективности воздействия в виде (54) необходимо знать условную величину рассматриваемого показателя эффективности – закон поражения.

Под законом поражения понимают зависимость между вероятностью поражения объекта и условиями, от которых это событие зависит.

В общем случае закон поражения – функция многих переменных: значений поражающих факторов, характеристик сопротивляемости и защищенности объекта, параметров внешних условий, сопровождающих воздействие, и других величин. Однако анализ попадания в конкретные объекты показывает, что только ограниченное число аргументов можно принимать во внимание, так как остальные оказывают малое влияние на величину вероятности попадания объекта. При рассмотрении эффективности воздействия группой факторов примем допущение о координатном законе поражения (КЗП) подэлемента СВ. Под координатным законом поражения будем понимать зависимость между вероятностью поражения отдельного объекта из состава неоднородной площади и координатами точки разрыва подэлемента СВ в системе ОХЗ, начало которой совпадает с центром отдельного объекта.

Примем допущение об аппроксимации КЗП подэлемента СВ показательным законом:

$$G(x_{lkrst}, z_{lkrst}) = S_{n_{lkrst}} \frac{\rho^2}{\pi E_{z_{lkrst}} E_{\phi_{lkrst}}} e^{-\rho^2 \left(\frac{x_{lkrst}^2}{E_{z_{lkrst}}^2} + \frac{z_{lkrst}^2}{E_{\phi_{lkrst}}^2} \right)}, \quad (55)$$

где $S_{n_{lkrst}}$ - приведенная площадь поражения объекта l-го подэлемента СВ k-го элемента, при организации процедур воздействия из r-го s-й группы СВ t-го фактора воздействия;

x_{lkrst}, z_{lkrst} – координаты точки распада l-го подэлемента СВ в системе, центр которой совпадает с центром отдельного объекта из состава неоднородной площади;

$E_{z_{lkrst}}, E_{\phi_{lkrst}}$ – параметры, определяемые по зависимости:

$$E_{\phi_{lkrst}} = \rho \sqrt{\frac{S_{n_{lkrst}}}{\pi G_0 \eta}}, \quad E_{z_{lkrst}} = \eta E_{\phi_{lkrst}},$$

где $\rho=0,4769$;

$$G_0 = G(0,0);$$

η – отношение приведенных размеров отдельного объекта из состава неоднородной площади.

Выражение (55) представляет собой условную вероятность поражения отдельного объекта при разрыве l-го подэлемента СВ k-го элемент СВ при организации процедур воздействия из r-го средства воздействия s-й группы СВ t-го фактор воздействия по i,j-й точке прицеливания. Полученная условная вероятность поражения отвечает фиксированным величинам ошибок подготовки и фиксированным ошибкам рассеивания.

Условная вероятность $p_{ijlkrst}(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}})$ - вероятность попадания объекта при распаде l-го подэлемента СВ k-го элемент СВ при организации процедур воздействия из r-й средство воздействия s-й группы СВ t-го фактор воздействия по i,j-й точке прицеливания, отвечающая всем возможным положениям разрыва при фиксированных ошибках подготовки и рассеивания точки вскрытия СВ, т. е.

$$\begin{aligned}
& p_{ijklrst} \left(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}} \right) = \\
& = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G \left(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}} \right) \cdot f \left(x_{\delta_{\partial lkrst}}, z_{\delta_{\partial lkrst}} / (x_n + x_{pc_{krst}}), (z_n + z_{pc_{krst}}) \right) dx_{\delta_{\partial lkrst}} dz_{\delta_{\partial lkrst}}
\end{aligned} \tag{56}$$

где $x_n = x_{\Gamma D} + x_{\Delta t} + x_{\delta_{st}} + x_{\text{БМ}_{rst}}$;

$z_n = z_{\Gamma D} + z_{\Delta t} + z_{\delta_{st}} + z_{\text{БМ}_{rst}}$;

$f \left(x_{\delta_{\partial lkrst}}, z_{\delta_{\partial lkrst}} / (x_n + x_{pc_{krst}}), (z_n + z_{pc_{krst}}) \right)$ – условная функция плотности отклонений распада подэлемента СВ от центра объекта при фиксированных ошибках подготовки и рассеивания точки вскрытия:

$$f \left(x_{\delta_{\partial lkrst}}, z_{\delta_{\partial lkrst}} / (x_n + x_{pc_{krst}}), (z_n + z_{pc_{krst}}) \right) = \frac{\rho^2}{\pi B \delta_{\partial lkrst} B \delta_{\partial lkrst}} e^{-\rho^2 \left[\frac{(x_{lkrst} - (\gamma_{x_{ij}} + x_n + x_{pc_{krst}}))^2}{B \delta_{\partial lkrst}^2} + \frac{(z_{lkrst} - (\gamma_{z_{ij}} + z_n + z_{pc_{krst}}))^2}{B \delta_{\partial lkrst}^2} \right]}$$

Формула (56) справедлива для любого КЗП.

Если подставить (54) в равенство (56), то, его можно преобразовать к виду

$$p_{ijklrst} \left(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}} \right) = S_{n_{lkrst}} \frac{\rho^2}{\pi B \delta_{\partial lkrst} B \delta_{\partial lkrst}} e^{-\rho^2 \left[\frac{(\gamma_{x_{ij}} + x_n + x_{pc_{krst}})^2}{B \delta_{\partial lkrst}^2} + \frac{(\gamma_{z_{ij}} + z_n + z_{pc_{krst}})^2}{B \delta_{\partial lkrst}^2} \right]}, \tag{57}$$

где $B \delta_{\partial lkrst} = \sqrt{B \delta_{\partial lkrst}^2 + E_{\partial lkrst}^2}$, $B \delta_{\partial lkrst} = \sqrt{B \delta_{\partial lkrst}^2 + E_{\partial lkrst}^2}$.

Таким образом, аппроксимация КЗП показательной функцией (54) позволяет преобразовать двойной интеграл (56) в функцию (57).

3.2 Показатель эффективности воздействия на противодействующую систему

Показатель эффективности воздействия на противодействующую систему рассмотрим применительно к группе факторов воздействия на элемент СВ по неоднородной площадной объекты

Ожидаемый результат воздействия по неоднородной площади характеризуется вероятностью ее поражения.

Если в районе отдельной объекта разорвалось несколько подэлемента СВ (в общем случае различного типа), то вероятность ее поражения определяется как вероятность того, что хотя бы один разрыв из них был удачным. Т.к. вероятность поражения объекта от разрыва к разрыву изменяется, то вероятность поражения объекта P рассчитывают по формуле

$$P = 1 - \prod_{l=1}^{N_{\sigma}} (1 - p_l)$$

где p_l – вероятность поражения объекта при разрыве одного подэлемента СВ.

Предположим, что по i, j -й точке присваивания выпущен k -й элемент из r -го элемента s -й группы СВ t -го фактора воздействия, включающей N_k подэлемента СВ при фиксированных групповых $(x_{гд}, z_{гд})$, фактор воздействия $(x_{дт}, z_{дт})$ и 1-я группа СВ ошибках подготовки $(x_{\sigma_{ст}}, z_{\sigma_{ст}})$.

Тогда в этом случае выражение для условной вероятности поражения объекта при организации процедур воздействия k -го элемента СВ, имеет вид:

$$P_{ijklrst}(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}}) = 1 - \prod_{l=1}^{N_k} [1 - p_{ijklrst}(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}})] \quad (58)$$

Учитывая, что ошибки рассеивания точки вскрытия СВ $x_{pc_{krst}}, z_{pc_{krst}}$ являются независимыми случайными непрерывными величинами, распределенными по нормальному закону, запишем выражение для полной вероятности поражения объекта:

$$\begin{aligned} P_{ijklrst}(\gamma_{x_{ij}} + x_n, \gamma_{z_{ij}} + z_n) &= \\ &= 1 - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(x_{pc_{krst}}, z_{pc_{krst}}) \prod_{l=1}^{N_k} [1 - p_{ijklrst}(\gamma_{x_{ij}} + x_n + x_{pc_{krst}}, \gamma_{z_{ij}} + z_n + z_{pc_{krst}})] dx_{pc_{krst}} dz_{pc_{krst}}, \end{aligned}$$

где $\varphi(x_{pc_{krst}}, z_{pc_{krst}})$ – нормальный закон распределения ошибок рассеивания точки вскрытия СВ:

$$\varphi(x_{pc_{krst}}, z_{pc_{krst}}) = \frac{\rho^2}{\pi B \delta_{pc_{krst}} B \bar{\delta}_{pc_{krst}}} e^{-\rho^2 \left[\frac{\left(x_{pc_{krst}} - (\gamma_{x_{ij}} + x_{z0} + x_{\delta_t} + x_{\bar{\delta}_{st}} + x_{BM_{rst}}) \right)^2}{B \delta_{pc_{krst}}^2} + \frac{\left(z_{pc_{krst}} - (\gamma_{z_{ij}} + z_{z0} + z_{\delta_t} + z_{\bar{\delta}_{st}} + z_{BM_{rst}}) \right)^2}{B \bar{\delta}_{pc_{krst}}^2} \right]}$$

где $B \delta_{pc_{krst}}$, $B \bar{\delta}_{pc_{krst}}$ - срединные отклонения ошибок рассеивания точки вскрытия СВ.

При воздействии r -й средства воздействия s -й 1-я группа СВ имеют место три группы ошибок: неповторяющиеся (ошибки рассеивания точек вскрытия СВ), неповторяющиеся (ошибки рассеивания подэлемента СВ относительно точки вскрытия СВ) и повторяющиеся (ошибки подготовки) $(x_{BM_{rst}}, z_{BM_{rst}})$. Ошибки подготовки определяют положение центра рассеивания точек вскрытия СВ, относительно которых происходит рассеивание подэлемента СВ и, следовательно, оказывают влияние на вероятность поражения отдельного объекта при распаде одного подэлемента СВ.

Предположим, что по объекту производится организация процедур воздействия из r -й средства воздействия s -й 1-я группа СВ t -го группы фактора воздействия с расходом N_r элемент СВ на объекты при фиксированных групповых $(x_{гд}, z_{гд})$, фактора воздействия $(x_{дт}, z_{дт})$ и 1-я группа СВ ошибках подготовки $(x_{\bar{\delta}_{st}}, z_{\bar{\delta}_{st}})$. Воздействие на объект проводится по нескольким точкам.

Выражение для условной вероятности поражения объекта при $N_{r_{ij}}$ организация процедур воздействия по i, j -й точке имеет вид:

$$P_{ijrst}(\gamma_{x_{ij}} + x_n, \gamma_{z_{ij}} + z_n) = 1 - \prod_{k=1}^{N_{r_{ij}}} [1 - p_{ijkrst}(\gamma_{x_{ij}} + x_n, \gamma_{z_{ij}} + z_n)] \quad (59)$$

При воздействии по любой точке с расходом $N_{r_{ij}}$ условная вероятность поражения объекта должна рассчитываться по (59). Тогда, воспользовавшись (58), можно определить вероятность поражения объекта при воздействии по всем точкам при неизменных ошибках подготовки:

$$P_{rst}(x_n, z_n) = 1 - \prod_{i=1}^{n_s} \prod_{j=1}^{n_\phi} \prod_{k=1}^{N_{ij}} \left[1 - p_{ijk rst}(\gamma_{x_{ij}} + x_n, \gamma_{z_{ij}} + z_n) \right],$$

где n_Γ – число объектов;

n_ϕ – число углов.

Учитывая, что ошибки подготовки $x_{BM_{rst}}, z_{BM_{rst}}$ являются независимыми случайными непрерывными величинами, распределенными по нормальному закону, запишем выражение для полной вероятности поражения объекта:

$$P_{rst} = 1 - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(x_{BM_{rst}}, z_{BM_{rst}}) \prod_{i=1}^{n_s} \prod_{j=1}^{n_\phi} \prod_{k=1}^{N_{ij}} \left[1 - p_{ijk rst}(\gamma_{x_{ij}} + x_n, \gamma_{z_{ij}} + z_n) \right] dx_{BM_{rst}} dz_{BM_{rst}},$$

где $\varphi(x_{BM_{rst}}, z_{BM_{rst}})$ – нормальный закон распределения ошибок подготовки средства воздействия:

$$\varphi(x_{BM_{rst}}, z_{BM_{rst}}) = \frac{\rho^2}{\pi E_{x_{BM_{rst}}} E_{z_{BM_{rst}}}} e^{-\rho^2 \left[\frac{(x_{BM_{rst}} - (\gamma_{x_{ij}} + x_{z\phi} + x_{\phi_t} + x_{\phi_{st}}))^2}{E_{x_{BM_{rst}}}^2} + \frac{(z_{BM_{rst}} - (\gamma_{z_{ij}} + z_{z\phi} + z_{\phi_t} + z_{\phi_{st}}))^2}{E_{z_{BM_{rst}}}^2} \right]},$$

где $E_{x_{BM_{rst}}}, E_{z_{BM_{rst}}}$ – срединные отклонения ошибок подготовки средства воздействия.

Рассмотрим воздействие s-й группы СВ в составе N_s средство воздействия t-й группы фактора воздействия по объекту с искусственным рассеиванием. Воздействие сопровождается четырьмя группами ошибок: неповторяющейся группой ошибок рассеивания подэлемента СВ на плоскости, неповторяющейся группой ошибок рассеивания точек вскрытия СВ и двумя группами повторяющихся ошибок подготовки: ошибок подготовки средство воздействия с функцией плотности $\varphi(x_{BM_{rst}}, z_{BM_{rst}})$ и 1-я группа СВ с функцией плотности $\varphi(x_{\phi_{st}}, z_{\phi_{st}})$. Также зафиксируем фактор воздействия ошибки подготовки (x_{ϕ_t}, z_{ϕ_t}) и ошибки подготовки группы фактора воздействия $(x_{\Gamma_d}, z_{\Gamma_d})$.

Рассуждая аналогично, получим зависимость для вероятности поражения объекта s-й группы СВ в составе N_s :

$$P_{st} = 1 - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(x_{\delta_{st}}, z_{\delta_{st}}) \prod_{r=1}^{N_s} [1 - P_{rst}] dx_{\delta_{st}} dz_{\delta_{st}},$$

где N_s – число средств воздействия в s-й группе СВ;

$\varphi(x_{\delta_{st}}, z_{\delta_{st}})$ – нормальный закон распределения 1-ой группы СВ ошибок подготовки, определяемый по зависимости:

$$\varphi(x_{\delta_{st}}, z_{\delta_{st}}) = \frac{\rho^2}{\pi E_{x_{\delta_{st}}} E_{z_{\delta_{st}}}} e^{-\rho^2 \left[\frac{(x_{\delta_{st}} - (\gamma_{xij} + x_{z\delta} + x_{\delta t}))^2}{E_{x_{\delta_{st}}}^2} + \frac{(z_{\delta_{st}} - (\gamma_{zij} + x_{z\delta} + x_{\delta t}))^2}{E_{z_{\delta_{st}}}^2} \right]},$$

где $E_{x_{\delta_{st}}}, E_{z_{\delta_{st}}}$ – срединные отклонения 1-ой группы СВ ошибок подготовки.

Рассмотрим влияние t-ой группы фактора воздействия по объектам с искусственным рассеиванием. Воздействие сопровождается пятью группами ошибок: неповторяющейся группой ошибок рассеивания подэлемента СВ на плоскости, неповторяющейся группой ошибок рассеивания точек вскрытия СВ и тремя группами повторяющихся ошибок подготовки: ошибок подготовки средство воздействия с функцией плотности $\varphi(x_{\text{БМ}_{\text{rst}}}, z_{\text{БМ}_{\text{rst}}})$, 1-я группа СВ с функцией плотности $\varphi(x_{\delta_{st}}, z_{\delta_{st}})$ и факторов воздействия ошибок подготовки $(x_{\text{дт}}, z_{\text{дт}})$. Также зафиксируем ошибки подготовки группы факторов воздействия $(x_{\text{гд}}, z_{\text{гд}})$ [6].

Рассуждая аналогично, получим зависимость для вероятности поражения объекта при t-м факторе воздействия:

$$P_t = 1 - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(x_{\delta_t}, z_{\delta_t}) \prod_{s=1}^{N_t} [1 - P_{st}] dx_{\delta_t} dz_{\delta_t},$$

где N_t – число 1-я группа СВ в t-м факторе воздействия;

$\varphi(x_{дт}, z_{дт})$ – нормальный закон распределения факторов воздействия ошибок подготовки, определяемый по зависимости:

$$\varphi(x_{дт}, z_{дт}) = \frac{\rho^2}{\pi E_{x_{дт}} E_{z_{дт}}} e^{-\rho^2 \left[\frac{(x_{дт} - (\gamma_{xij} + x_{зд}))^2}{E_{x_{дт}}^2} + \frac{(z_{дт} - (\gamma_{zij} + x_{зд}))^2}{E_{z_{дт}}^2} \right]},$$

где $E_{x_{дт}}, E_{z_{дт}}$ – срединные отклонения факторов воздействия ошибок подготовки.

При воздействии группой факторов добавляется еще одна группа ошибок, повторяющаяся для всех организация процедур группы факторов воздействия: групповые ошибки подготовки по дальности $x_{гд}$ и групповые ошибки подготовки по направлению $z_{гд}$. Ошибки подготовки по дальности и направлению образуют систему независимых случайных величин $x_{гд}, z_{гд}$, которая характеризуется функцией плотности распределения $\varphi(x_{гд}, z_{гд})$.

Рассуждая аналогично, запишем формулу для расчета вероятности поражения объекта при воздействии всей группой факторов воздействия:

$$P = 1 - \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \varphi(x_{зд}, z_{зд}) \prod_{t=1}^K [1 - P_t] dx_{зд} dz_{зд},$$

где K – число факторов воздействия в группе;

$\varphi(x_{гд}, z_{гд})$ – нормальный закон распределения групповых ошибок подготовки, определяемый по зависимости:

$$\varphi(x_{зд}, z_{зд}) = \frac{\rho^2}{\pi E_{x_{зд}} E_{z_{зд}}} e^{-\rho^2 \left[\frac{(x_{зд} - \gamma_{xij})^2}{E_{x_{зд}}^2} + \frac{(z_{зд} - \gamma_{zij})^2}{E_{z_{зд}}^2} \right]}, \quad (60)$$

где $E_{x_{гд}}, E_{z_{гд}}$ – срединные отклонения групповых ошибок подготовки.

Таким образом, получена целевая функция для последующего решения задачи оптимизации состава привлекаемых боевых машин элемент СВ и

способа воздействия неоднородной площади в виде формульной зависимости (60).

Для практического использования полученных формульных зависимостей необходимо от неопределенных пределов интегрирования перейти в конечным пределам интегрирования. Вместо зависимости (60) будем использовать следующую зависимость [7]:

$$P = 1 - \int_{-5E_{x_{20}}}^{5E_{x_{20}}} \int_{-5E_{z_{20}}}^{5E_{z_{20}}} \varphi(x_{20}, z_{20}) \prod_{t=1}^K [1 - P_t] dx_{20} dz_{20}. \quad (61)$$

Также во всех формулах с неопределенными пределами произведем замену на пределы от $-5E_i$ до $5E_i$, где i – индекс соответствующей ошибки подготовки.

Для определения величины интеграла (61) можно выбрать удобный численный метод интегрирования. Например, обстоятельствами, влияющими на выбор метода, могут быть:

- простота программирования задачи;
- машинное время решения одного варианта;
- загрузка памяти ЭВМ;
- простота оценки погрешности метода;
- применимость к широкому кругу задач.

В частности, для определения вероятности поражения объекта можно использовать метод имитационного моделирования.

3.3 Алгоритм расчета кратных интегралов методом статистических испытаний Монте-Карло

В данном подразделе рассматривается теоретическая часть построения алгоритмов расчета кратных интегралов методом статистических испытаний (Монте-Карло).

Альтернативным от рассмотренного метода определения вероятности воздействия по противодействующей системе является использование метода имитационного моделирования (метода статистических испытаний), который

при наличии быстродействующих ЭВМ позволяет решить задачу без существенных допущений. При этом обработка результатов реализаций даже не требуется, а каждый результат можно непосредственно использовать для последующих расчетов. Точность метода лимитируется не его сущностью, а точностью исходных данных и числом проведенных реализаций. Реализация метода статистических испытаний основывается на задании следующих исходных данных:

- совместные характеристики уязвимости противодействующей системы и могущества средства воздействия (для обычных средств воздействия данные характеристики выражаются законами воздействия и формализуются значением приведенной площади воздействия);

- характеристики системы ошибок, сопровождающих воздействие (выражаются понятиями меткости, кучности, точности воздействия и формализуются соответствующими срединными отклонениями ошибок), которые, как правило, принимаются в виде случайных величин ошибок воздействия по дальности и по направлению, распределенным по независимым нормальным законам;

- параметры способа воздействия по противодействующей системе (формализуются количеством параметров способа воздействия).

Алгоритм применения метода статистических испытаний представлен в виде UML-диаграммы деятельности на рисунках 13, 14. Диаграммы деятельности/активностей (Activity Diagrams) являются представлением алгоритмов некоторых действий (активностей), выполняющихся в системе.

Диаграммы деятельности позволяют описать логику процедур, бизнес-процессы и потоки работ. Во многих случаях они напоминают схемы алгоритмов и программ, но принципиальная разница между диаграммами деятельности и нотацией схем алгоритмов и программ заключается в том, что диаграммы поддерживают параллельные процессы.

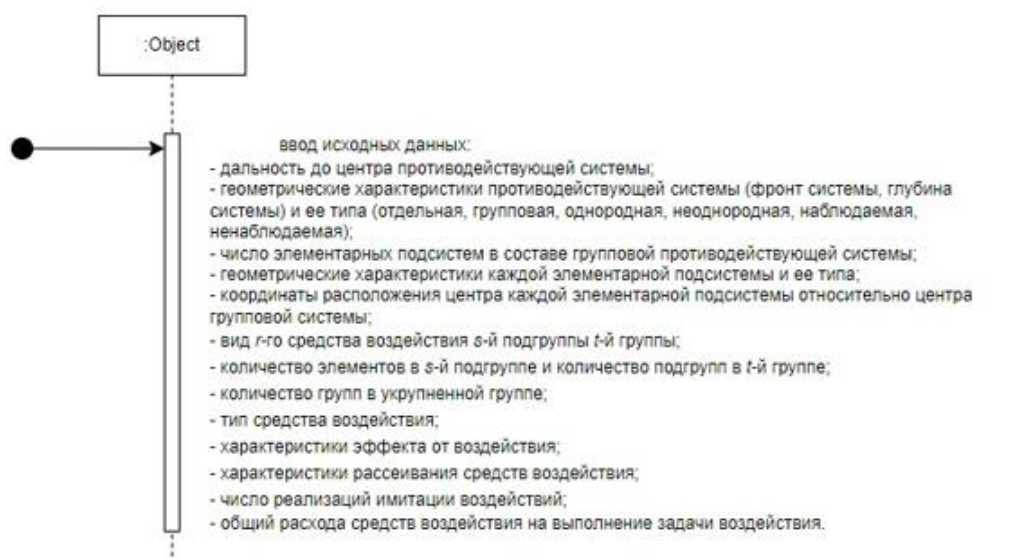


Рисунок 13 – Метод статистических испытаний.

UML-диаграмма деятельности

Работа алгоритма представляет собой последовательный вычислительный процесс, в котором можно выделить ряд этапов.

Этап 1. На данном этапе осуществляется ввод следующих исходных данных:

- дальности до центра противодействующей системы;
- геометрических характеристик противодействующей системы (фронт системы, глубина системы) и ее типа (отдельная, групповая, однородная, неоднородная, наблюдаемая, ненаблюдаемая);
- числа элементарных подсистем в составе групповой противодействующей системы;
- геометрических характеристики каждой элементарной подсистемы и ее типа;
- координат расположения центра каждой элементарной подсистемы относительно центра групповой системы;
- вида r -го средства воздействия S -й подгруппы t -й группы;
- количества элементов в S -й подгруппе и количество подгрупп в t -й группе;
- количества групп в укрупненной группе;

- типа средства воздействия;
- характеристик эффекта от воздействия;
- характеристик рассеивания средств воздействия;
- числа реализаций имитации воздействий;
- общего расхода средств воздействия на выполнение задачи воздействия.

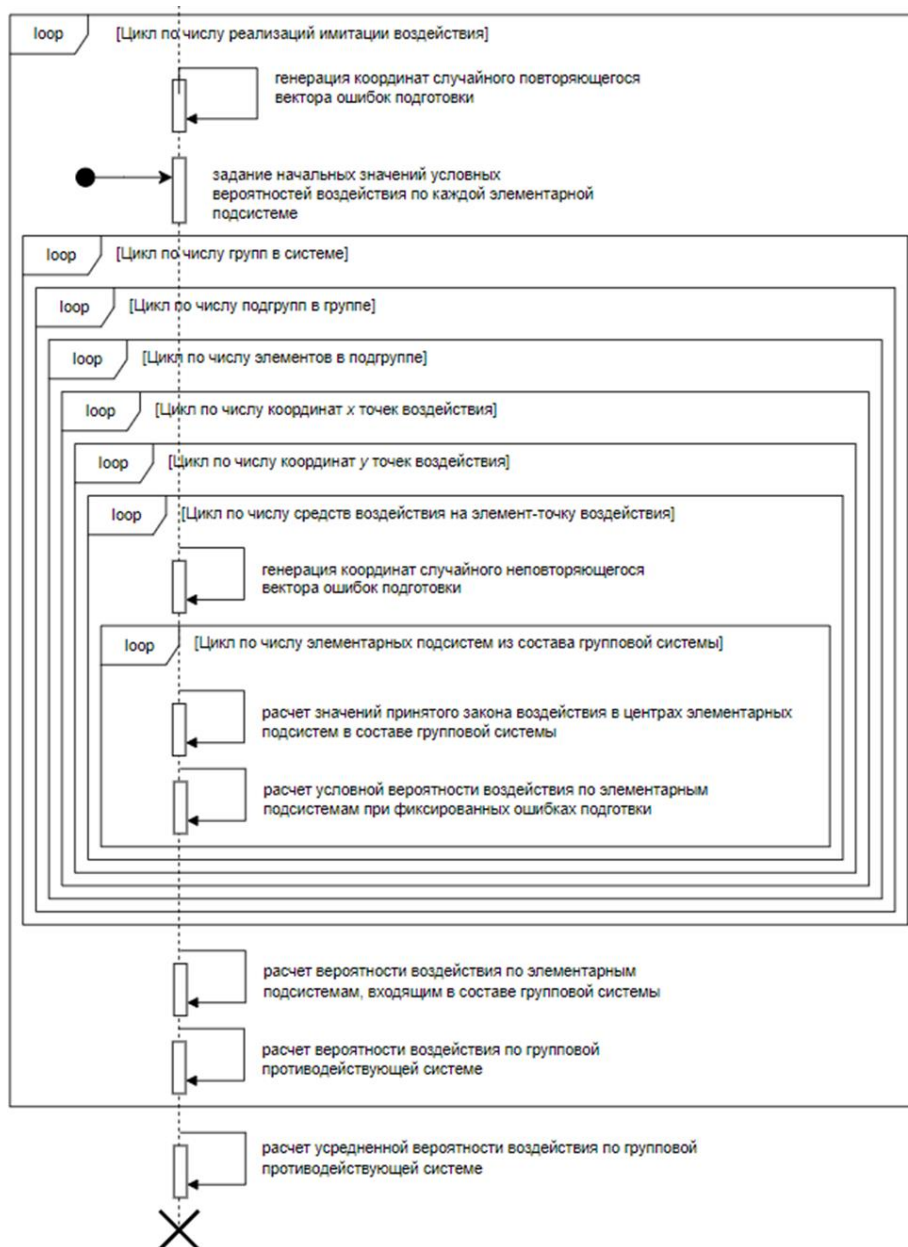


Рисунок 14 - Метод статистических испытаний.

UML-диаграмма деятельности (продолжение)

Этап 2. На данном этапе производится расчет параметров способа воздействия: числа точек воздействия, расхода средств воздействия на элемент-точку воздействия.

Этап 3. На данном этапе производится расчет элементных, подгрупповых, групповых и ошибок подготовки системы.

Этап 4. На данном этапе определяются координаты центров элементарных подсистем в относительно центра групповой противодействующей системы.

Этап 5. На данном этапе производится непосредственное вычисление условных вероятностей воздействия по элементарным подсистемам из состава групповой системы с использованием следующего ряда циклических процедур:

- цикл № 1. Цикл по числу реализаций имитации воздействия;
- цикл № 2. Цикл по числу групп в системе;
- цикл № 3. Цикл по числу подгрупп в группе;
- цикл № 4. Цикл по числу элементов в подгруппе;
- цикл № 5. Цикл по числу координат x точек воздействия;
- цикл № 6. Цикл по числу координат y точек воздействия;
- цикл № 7. Цикл по числу средств воздействия на элемент - точку воздействия;
- цикл № 8. Цикл по числу элементарных подсистем из состава групповой противодействующей системы.

Цикл № 1 по числу реализаций имитации воздействия является внешним для всех остальных циклов. Внутри этого цикла производится генерация координат случайного повторяющегося вектора ошибок подготовки воздействия элементом, подгруппой, группой и всей системой.

С помощью датчика случайных чисел получают проекции случайного вектора ошибок подготовки, повторяющегося для всех реализаций воздействия элементов всех подгрупп и групп системы (ошибки системы по дальности и по направлению), по следующим формулам:

$$x^{\partial} = N\left(\gamma_x^{(i,j)}, E_x^{\partial}\right), \quad z^{\partial} = N\left(\gamma_z^{(i,j)}, E_z^{\partial}\right), \quad (62)$$

где $N(m,E)$ – случайные числа, распределенные по нормальному закону с математическим ожиданием m и средним отклонением E .

Используя результаты расчетов по формулам (62), с помощью датчика случайных чисел получают проекции случайного вектора ошибок подготовки, повторяющегося для всех реализаций воздействий элементов t -й группы системы по дальности и по направлению, по следующим формулам:

$$x_t^{\partial} = N\left(x^{\partial}, E_{x_t}^{\partial}\right), \quad z_t^{\partial} = N\left(z^{\partial}, E_{z_t}^{\partial}\right). \quad (63)$$

Используя результаты расчетов по формулам (63), с помощью датчика случайных чисел получают проекции случайного вектора ошибок подготовки воздействия, повторяющегося для всех реализаций воздействия S -й подгруппы t -й группы системы по дальности и по направлению по следующим формулам:

$$x_{st}^{\partial} = N\left(x_t^{\partial}, E_{x_{st}}^{\partial}\right), \quad z_{st}^{\partial} = N\left(z_t^{\partial}, E_{z_{st}}^{\partial}\right). \quad (64)$$

Используя результаты расчетов по формулам (64), с помощью датчика случайных чисел получают проекции случайного вектора ошибок подготовки, повторяющегося для всех реализаций воздействия $г$ -го средства воздействия S -й подгруппы t -й группы системы по дальности и по направлению по следующим формулам:

$$x_{rst}^{op} = N\left(x_{st}^{\partial}, E_{x_{rst}}^{op}\right), \quad z_{rst}^{op} = N\left(z_{st}^{\partial}, E_{z_{rst}}^{op}\right). \quad (65)$$

Кроме генерации случайных чисел в этом цикле производится задание начальных значений условных вероятностей воздействия по элементарным подсистемам. Задавая значения вероятностей отличные от нуля, подразумевают, что противодействующая подсистема уже подвергалась ранее воздействию. Для подсистемы, ранее не подвергшейся воздействию эти значения предполагаются нулевыми.

В целом внутри циклов № 2...№ 7 производится генерация координат случайного неповторяющегося вектора ошибок рассеивания.

Далее, используя результаты расчетов по формулам (65), с помощью датчика случайных чисел получают проекции случайного вектора ошибок рассеивания средств воздействия, неповторяющегося для каждой k -й реализации воздействия r -го средства воздействия S -й подгруппы t -й группы системы по дальности и по направлению по следующим формулам:

$$x_{krst}^c = N(x_{rst}^{op}, B_{\partial k rst}^c), \quad z_{krst}^c = N(z_{rst}^{op}, B_{\bar{o} k rst}^c).$$

Далее в цикле № 8 производится расчет условной вероятности воздействия по элементарным подсистемам, при фиксированных ошибках подготовки и рассеивания по следующим рекуррентным соотношениям:

$$\begin{cases} p_m^{(0)} = 1 - (1 - G^{(0)}), \quad l = 0, \\ p_m^{(l)} = 1 - (1 - G^{(l-1)})(1 - G^{(l)}), \quad l = 1 \dots N_{ij}, \end{cases}$$

где $p_m^{(0)}$ – начальное значение условной вероятности воздействия по m -й элементарной подсистеме из состава групповой противодействующей системы;

$p_m^{(l)}$ – значение условной вероятности воздействия по m -й элементарной подсистеме из состава групповой противодействующей системы при l -м срабатывании подэлемента относительно (i,j) -ой точки прицеливания;

$G^{(0)}$ – начальное значение условной вероятности воздействия по элементарной подсистеме, в общем случае отличное от нуля;

$G^{(l)}$ – значение условной вероятности воздействия по элементарной подсистеме из состава групповой противодействующей системы, при l -м срабатывании средства воздействия, отвечающая фиксированным ошибкам подготовки и рассеивания средств воздействия.

На следующем этапе производится расчет вероятности воздействия по элементарным подсистемам, входящих в состав групповой противодействующей системы по формуле:

$$P_{c_m} = p_m^{(N_{ij})}, \quad m = 1 \dots N_c,$$

где P_{c_m} – вероятность воздействия по m -й элементарной подсистеме из состава групповой противодействующей системы.

На следующем этапе производится расчет вероятности воздействия по групповой противодействующей системе на u -й реализации имитации воздействия по формуле:

$$P^{(u)} = \sum_{m=1}^{N_c} \alpha_m P_{c_m},$$

где α_m – коэффициент важности m -й элементарной подсистемы из состава групповой противодействующей системы.

На заключительном этапе алгоритма производится расчет вероятности воздействия по групповой противодействующей системе, методом усреднения по числу реализаций имитации воздействия:

$$P_p = \frac{\sum_{u=1}^U P^{(u)}}{U},$$

где U – заданное число реализаций в методе статистических испытаний.

3.4 Программного обеспечения расчета кратных интегралов методом статистических испытаний

На основании материалов, изложенных выше, было разработано программное обеспечение расчета кратных интегралов методом статистических испытаний применительно к стрельбе группы дивизионов РСЗО по неоднородной площади объекта.

Для реализации алгоритма расчета кратных интегралов методом статистических испытаний выбрана интегрированная система компьютерной математики MATLAB.

Функция расчета вероятности успешного воздействия по противодействующей системе методом статистических испытаний, составляющая основную часть программы и предназначенная для вычисления

кратных интегралов представлена ниже (листинг 1). Полностью листинг программы приведен в приложении Б.

Листинг 1 – Функция расчета вероятности успешного воздействия по противодействующей системе методом статистических испытаний

```
function y = vsf(Pz,FR,GL,Fr,Gl,Kgr,Vk,ng,nf,Sn,X,Z,xtp,ztp,G0,n,X2,nr,nc,a,Vog)
% расчет вероятности успешного воздействия по противодействующей системе методом
% имитационного моделирования (расчет кратных интегралов)
%-----

%-----
% формирование матриц признаков наличия элементов воздействия в группах
XX3 = X2(sum(Vk)*ng*nf+1:end);
for k=1:1:Kgr
    OS{k} = XX3(sum(Vk(1:k-1))+1:sum(Vk(1:k)));
end
%-----
% формирование матриц расхода средств воздействия для каждого элемента
% воздействия по всем точкам
% воздействия
XX2 = X2(1:sum(Vk)*ng*nf);
N = reshape(XX2,[length(xtp),length(ztp),sum(Vk)]);
%-----
for k=1:1:Kgr
    N1{k} = N(:, :, sum(Vk(1:k-1))+1:sum(Vk(1:k)));
end

for k=1:1:Kgr
    for v=1:1:Vk(k)
        N1{k}(:, :, v) = OS{k}(v).*N1{k}(:, :, v);
    end
end

%-----
global PP; % относительные координаты размещения осей в пикселях
(относительно размеров figure)
P3 = zeros(1,nr); % выделение памяти для массива вероятностей успешности
воздействия
% размеры отдельных элементов из состава противодействующей системы по фронту и
глубине в
% пикселях
% (при nc=1 - размеры отдельного элемента)
if nc ~= 1
    for i=1:1:nc
        fr{i} = ((PP(3)-PP(1))/2+Z(i)-Fr(i)/2):1:((PP(3)-PP(1))/2+Z(i)+Fr(i)/2-1); % фронт элемента
        gl{i} = ((PP(4)-PP(2))/2+X(i)-Gl(i)/2):1:((PP(4)-PP(2))/2+X(i)+Gl(i)/2-1); % глубина элемента
    end
else
    fr{1} = ((PP(3)-PP(1))/2-FR/2):1:((PP(3)-PP(1))/2+FR/2-1); % фронт элемента
    gl{1} = ((PP(4)-PP(2))/2-GL/2):1:((PP(4)-PP(2))/2+GL/2-1); % глубина
    элемента
end
%-----
% координаты центра противодействующей системы
ZZZ = (PP(3)-PP(1))/2;
XXX = (PP(4)-PP(2))/2;
```

```

for ii=1:1:nr % начало цикла реализаций
%-----
% генерация координат случайного вектора системной ошибки подготовки
z0 = normrnd(ZZZ,Vog.Ez(3)/0.6745);
x0 = normrnd(XXX,Vog.Ex(3)/0.6745);
% генерация координат случайного вектора групповой ошибки подготовки
z1 = zeros(1,Kgr);
x1 = zeros(1,Kgr);
for k=1:1:Kgr
    z1(k) = normrnd(z0,Vog.Ez(2)/0.6745);
    x1(k) = normrnd(x0,Vog.Ex(2)/0.6745);
end

for k=1:1:Kgr
    for v=1:1:Vk(k)
        z2(k,v) = normrnd(z1(k),Vog.Ez(1)/0.6745);
        x2(k,v) = normrnd(x1(k),Vog.Ex(1)/0.6745);
    end
end

% задание начальных значений КЗП по противодействующей системе (отдельным
элементам)
G1 = zeros(1,nc);
if nc ~= 1
    for i=1:1:nc
        G1(i) = 0; % по каждого отдельного элемента
    end
else
    G1(1) = 0; % по всей противодействующей системе
end

%-----
% задание пустого массива условных вероятностей успешности воздействия
Pr = [];
%-----
% непосредственно расчет вероятности успешности воздействия
for i2=1:1:Kgr % цикл по числу групп в системе воздействия
    for v=1:1:Vk(i2) % цикл по числу элементов воздействия в каждой группе
        if OS{i2}(v) ~= 0 %
            for j=1:1:length(xtp) % цикл по числу установок прицела
                for k=1:1:length(ztp) % цикл по числу установок угломера
                    if N1{i2}(j,k,v) ~= 0
                        for l=1:1:N1{i2}(j,k,v) % цикл по количеству средств
воздействия на каждую точку воздействия
                            % генерация координат точки падения
                            % средства воздействия
                            z3(i2,v,j,k,l) =
normrnd(z2(i2,v)+ztp(k),Vog.Vb/0.6745);
                            x3(i2,v,j,k,l) =
normrnd(x2(i2,v)+xtp(j),Vog.Vd/0.6745);
                            %-----
                            for i1=1:1:nc % цикл по числу отдельных
элементам противодействующей системы
                                % пересчет координат точек элемента
                                % относительно точки падения средства
                                % воздействия
                                x11(i1) = (PP(4)-PP(2))/2+X(i1)-
x3(i2,v,j,k,l);
                                z11(i1) = (PP(3)-PP(1))/2+Z(i1)-
z3(i2,v,j,k,l);
                                % расчет КЗП в точках элемента
                                G2(i1) =
KZPEXP(G0(i1),Sn(i1),n(i1),x11(i1),z11(i1));

```

```

-----
%-----
Pr(i1) = 1-(1-G2(i1)).*(1-G1(i1)); %
условная вероятность успешности воздействия при фиксированных ошибках подготовки
G1(i1) = Pr(i1);
end
end
end
end
end
end
end
end
end
end
%-----
% вероятность успешности воздействия по отдельным элементам
% противодействующей системы
if isempty(Pr) == 0
    if nc ~= 1
        for i=1:1:nc
            P2(ii,i) = Pr(i);
        end
        % вероятность успешности воздействия по противодействующей
        % системе
    else
        P2(ii) = Pr(1);
    end
else
    P2 = zeros(ii,nc);
end
end
%-----
end % конец цикла по числу реализаций

if nr == 1
    P = sum(a.*P2);
else
    P11 = mean(P2);
    P = sum(a.*P11);
end

y = -P;

function [G] = KZPEXP(G0,Sn,n,x,z)
% расчет координатного закона воздействия (показательная аппроксимация)
%-----
% Sn - приведенная площадь воздействия элемента
% n - параметр закона воздействия (отношение срединных отклонений)
% G0 - параметр закона воздействия средства воздействия в точке (0,0)
% x - координата x точки действия средства воздействия
% z - координата z точки действия средства воздействия
ro = 0.47694;
Ef = ro*sqrt(Sn/(pi*G0*n));
Eg = n*Ef;
G = Sn*ro^2/(pi*Eg*Ef)*exp(-ro^2*(x^2/Eg^2+z^2/Ef^2));

```

Результаты расчетов усредненной вероятности воздействия по групповой противодействующей системе, полученные в ходе многократного прогона задачи на ЭВМ при различных значениях числа реализаций имитации случайных воздействий сведены в таблицу 5.

Таблица 5 – Результаты моделирования

Число реализаций имитации случайных воздействий	Усредненная вероятность воздействия по групповой противодействующей системе
50	0,453
100	0,622
150	0,723
200	0,739
250	0,756
300	0,804
350	0,877

Анализ результатов показывает, что с увеличением числа реализаций имитации случайных воздействий усредненная вероятность воздействия по групповой противодействующей системе увеличивается.

Заключение

В первом разделе были рассмотрены следующие численные методы для приближенного вычисления кратных интегралов: методы расчета кратных интегралов; численные методы вычисления двойного интеграла и характеристика метода статистических испытаний (Монте-Карло). В результате рассмотрения и применения данных методов была поставлена задача разработки ПО.

Во втором разделе рассмотрена сущность метода Монте-Карло и моделирование случайных величин. В ходе работы были изучены следующие алгоритмы: общий алгоритм расчета кратных интегралов методом Монте-Карло, алгоритмы вычисления общих интегралов и кратных интегралов.

В третьем разделе было реализовано программное обеспечение расчета кратных интегралов методом статистических испытаний при расчете вероятности воздействия на противодействующую систему. Для этого были рассмотрены: характеристика системы ошибок при воздействии на противодействующую систему и показатель эффективности воздействия на противодействующую систему. В следующем подразделе «алгоритм расчета кратных интегралов методом статистических испытаний Монте-Карло» рассмотрена теоретическая часть построения алгоритмов расчета кратных интегралов методом статистических испытаний (Монте-Карло).

На основании собранных материалов, было разработано программное обеспечение расчета кратных интегралов методом статистических испытаний применительно к стрельбе группы дивизионов РСЗО по неоднородной площади объекта.

Для реализации алгоритма расчета кратных интегралов методом статистических испытаний выбрана интегрированная система компьютерной математики MATLAB.

В результате выполнения выпускной квалификационной работы рассмотрено и создано программное обеспечение для вычисления кратных

интегралов методом статистических испытаний, в основе которой лежит генерация случайных чисел.

Список использованных источников

1. Абчук В.А., Матвейчук Ф.А., Томашевский Л.П. Справочник по исследованию операций. – М.: ВИ МО СССР, 1979.
2. Бендат Дж., Пирсол А. Измерение и анализ случайных процессов. – М.: Мир, 1971.
3. Бочаров П.П., Печенкин А.В. Теория массового обслуживания. – М.: Из-во РУДН, 1995.
4. Бусленко Н.П., Голенко Д.И., Соболев И.М., Срагович В.Г., Шрейдер Ю.А. Метод статистических испытаний (метод Монте-Карло). – М.: Физматгиз, 1962.
5. Бусленко Н.П. Моделирование сложных систем. – М.: Наука, 1978.
6. Вентцель Е.С. Исследование операций. – М.: Сов. радио, 1972.
7. Вентцель Е.С. Исследование операций. Задачи, принципы, методология. – М.: Гл.ред. физ. мат. лит., 1988.
8. Вентцель Е.С. Теория вероятностей. – М.: Наука, 1969.
9. Гмурман В.Е. Теория вероятностей и математическая статистика. – М.: Высшая школа, 1977.
10. Голенко Д.И. Моделирование и статистический анализ псевдослучайных чисел на ЭВМ. – М.: Наука, 1965.
11. Громов В.А., Левашов В.Ф. Основы статистического моделирования. Ч. III. Обработка результатов экспериментов. – Пенза.: ВАИУ, 1997.

Приложение А
ГЛОССАРИЙ
(справочное)

ММК (Методы Монте-Карло) – группа численных методов для изучения случайных процессов.

СУБД (Система управления базой данных) – специализированное программное обеспечение, предоставляющее программный интерфейс, который дает возможность создавать, модифицировать и использовать базу данных.

HTTP (HyperText Transfer Protocol) – протокол прикладного уровня передачи данных, основанный на технологии «клиент-сервер».

UML (Unified Modeling Language) – язык графического описания для объектного моделирования в области разработки программного обеспечения, моделирования бизнес-процессов, системного проектирования и отображения организационных структур.

SQL (Structured Query Language) – декларативный язык программирования, применяемый для создания и модификации данных в реляционной базе данных.

ООП – объектно ориентированное программирование, методология программирования, основанная на представлении программы в виде совокупности объектов.

Приложение Б
ИСХОДНЫЙ КОД ПРОГРАММЫ
(рекомендуемое)

Листинг Б.1 – Функция вычисления кратных интегралов методом Монте – Карло

```
function kratinteg %#codegen
% задача расчета кратных интегралов (применительно к системе воздействия)

tic,
clearvars;
global h; % задание массива дескрипторов
global scrsz; % задание размеров экрана монитора
global PP; % относительные координаты размещения осей в пикселях
(относительно размеров figure)
global Pos1;
global Vog;

h = zeros(9000,1); % задание нулевого массива (9000 строк 1 столбец)
%-----
scrsz = get(0, 'ScreenSize'); % получение размеров экрана монитора
%%-----
% создание главного окна-----
%-----
Pos1 = [scrsz(1)+5 scrsz(2)+35 scrsz(3)-10 scrsz(4)-90];
h(20) = figure('MenuBar','None',...
               'Name','Задача расчета кратных интегралов (применительно к
системе воздействия)',...
               'NumberTitle','Off',...
               'Position',Pos1);
Pos = [0.25 0.05 0.73 0.85];
% координаты осей в пикселях
PP(1) = floor(Pos(1)*(Pos1(3)-Pos1(1)));
PP(2) = floor(Pos(2)*(Pos1(4)-Pos1(2)));
PP(3) = floor(Pos(3)*(Pos1(3)-Pos1(1)));
PP(4) = floor(Pos(4)*(Pos1(4)-Pos1(2)));
%-----
Vog.Dcc = 10000; % Дальность до противодействующей системы
Vog.dTz = 0; % параметр ветвления
%-----
% задание исходных данных по противодействующей системе
%-----
% геометрические характеристики отдельных элементов из состава
противодействующей системы
% (если противодействующая система не групповая - все значения равны 0)
lx = [2.45,3.20,3.20,3.05,3.05,3.05,3.05,3.30,3.30,3.30,3.30,2.45,3.05,3.05,3.05,3.05,
3.30,3.30,3.30,3.30,...
2.45,5.85,5.85,5.85,5.85]; % половина глубины отдельных элементов
lz = [1.35,1.20,1.20,1.95,1.95,1.95,1.95,1.55,1.55,1.55,1.55,1.35,1.95,1.95,1.95,1.95,
1.55,1.55,1.55,1.55,...
1.35,1.20,1.20,1.20,1.20]; % половина фронта отдельных элементов

Gl = 2*lx; % глубина отдельных элементов
Fr = 2*lz; % фронт отдельных элементов

% координата X центра отдельного элемента относительно центра противодействующей
системы
Xc = [205,180,150,420,475,485,465,340,390,415,395,300,445,485,475,440,375,...
415,410,365,305,50,50,50,50];
Xcv = Xc + lx;
Xcn = Xc - lx;
GL = max(Xcv)+min(Xcn); % глубина противодействующей системы
Lx = GL/2; % половина глубины противодействующей системы
Xc = Lx - Xc;
```

```

% координата Z центра отдельного элемента относительно центра противодействующей
системы
Zc = [340,295,380,20,90,195,290,20,110,215,290,130,415,510,610,690,430,...
      515,585,680,545,15,45,110,175];
Zcv = Zc + lz;
Zcn = Zc - lz;
FR = max(Zcv)+min(Zcn); % фронт противодействующей системы
Lz = FR/2; % половина фронта противодействующей системы
Zc = Lz - Zc;
nc = length(Xc); % число отдельных элементов из состава противодействующей
системы
% важность отдельных элементов из состава противодействующей системы
a = [0.012,0.012,0.012,0.1,0.1,0.1,0.1,0.012,0.012,0.012,0.012,0.012,0.1,0.1,...
     0.1,0.1,0.012,0.012,0.012,0.012,0.012,0.012,0.012,0.012,0.012];
%-----
% рисование элементов
% задание равномерной прямоугольной сетки точек воздействия
nf = 10; % число столбцов сетки
ng = 10; % число строк сетки

% определение координат точек воздействия
XX = zeros(1,ng);
ZZ = zeros(1,nf);
dx2 = GL/(ng-1);
dz2 = FR/(nf-1);
for i=1:1:ng
    XX(i) = -GL/2 + dx2*(i-1);
end
for j=1:1:nf
    ZZ(j) = -FR/2 + dz2*(j-1);
end
%-----
figure(1); % установка текущего окна
axes('XLim', [(-2*FR) (4*FR)], 'YLim', [(-2*GL) (4*GL)]); % установка осей под
размеры объекта
cla; % очистка осей
hold on; % включение наложения
grid on; % включение сетки
if nc ~= 1
    for i=1:1:nc
        rectangle('Position', [(PP(3)-PP(1))/2+Zc(i)-Fr(i)/2, (PP(4)-
PP(2))/2+Xc(i)-Gl(i)/2, Fr(i), Gl(i)], 'Curvature', [0,0]);
    end
end
% рисование границы противодействующей системы (при nc=1 - отдельного элемента)
rectangle('Position', [(PP(3)-PP(1))/2-FR/2, (PP(4)-PP(2))/2-
GL/2, FR, GL], 'Curvature', [0,0]);
% рисование исходных точек воздействия
for i=1:1:ng
    for j=1:1:nf
        plot((PP(3)-PP(1))/2+ZZ(j), (PP(4)-
PP(2))/2+XX(i), 'rs', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize', 6);
    end
end
%-----
% задание исходных данных по составу системы воздействия
%-----
% задание признаков наличия элементов системы воздействия по группам
XXX{1} = [1 1 1]; % элементы в 1-й группе
XXX{2} = [1 1 1]; % элементы во 2-й группе
% XXX{3} = [1 1 1 1 1 1 1 1]; % элементы в 3-й группе
% XXX{4} = [1 1 1 1 1 1 1 1]; % элементы в 4-й группе
% XXX{5} = [1 1 1 1 1 1 1 1]; % элементы в 5-й группе

```

```

% XXX{6} = [1 1 1 1 1 1 1 1]; % элементы в 6-й группе
%-----
Kgr = length(XXX); % число групп
Vk = zeros(1,Kgr);
for k=1:1:Kgr
    Vk(k) = sum(XXX{k}); % число однотипных элементов системы воздействия в
    группах
end

%-----
X1 = cat(2,XXX{1:end}); % признак наличия элемента системы воздействия

for k=1:1:Kgr
    for j=1:1:Vk(k)
        rs{k,j} = zeros(nf,ng);
        B{k,j} = reshape(rs{k,j},[1,ng*nf]);
        Bk{k,j} = 40; % количество средств воздействия в элементе воздействия
    end
end

BK = cat(2,Bk{1:end}); % количество средств воздействия у каждого элемента
воздействия
X2 = cat(2,B{1:end}); % расход средств воздействия для каждого элемента
воздействия по всем точкам воздействия
%-----

%-----
% задание исходных данных по элементу воздействия
%-----
% число средств воздействия, расходуемых за минуту
lambda = 120;
%-----
% задание исходных данных по средству воздействия
%-----
% характеристики действия средства воздействия по отдельному элементу из
% состава противодействующей системы
% (размерность массивов - nc)
% приведенные зоны воздействия для различных типов элементов
% тип одиночного элемента из состава противодействующей системы
% 1 - тип 1
% 2 - тип 2
% 3 - тип 3
% 4 - тип 4
% 5 - тип 5
% 6 - тип 6
% 7 - тип 7
% 8 - тип 8
% 9 - тип 9
% 10 - тип 10
% 11 - тип 11
% 12 - тип 12
Tip = [8,9,9,3,3,3,3,3,3,3,8,3,3,3,3,3,3,3,8,9,9,9,9];
Sn = zeros(1,length(Tip)); % выделение памяти под массив Snp
for i=1:1:length(Tip)
    switch Tip(i)
        case 1
            D = [4000 8000 12000 16000 20000];
            SPR = [355 370 390 430 500];
            Sn(i) = interp1(D,SPR,Vog.Dcc,'spline');
        case 2
            Sn(i) = 22;
        case 3
            D = [4000 8000 12000 16000 20000];
            SPR = [85 50 50 50 45];

```



```

1 0 0 1 0 0 0; % группа групп элементов воздействия
0 0 0 0 0 0 0; % группа элементов воздействия
0 1 1 0 1 1 1]; % элемент воздействия

% ошибки подготовки-----
% ошибки определения координат и высоты элемента
Vog.Excc = 10; % Срединная ошибка в координатах элемента по дальности, м
Vog.Ezcc = 10; % Срединная ошибка в координатах элемента по направлению, м
Vog.Ehcc = 3; % Срединная ошибка в координатах элемента по высоте, м
% ошибки топогеодезической подготовки
Vog.Expp = 10; % Срединная ошибка в координатах ОП по дальности, м
Vog.Ezpp = 10; % Срединная ошибка в координатах ОП по направлению, м
Vog.Ehpp = 2; % Срединная ошибка в координатах ОП по высоте, м
Vog.Eosn = 1; % Срединная ошибка ориентирования элемента воздействия в
основном направлении, тыс.
% ошибки баллистической подготовки
Vog.Exmz = 2.0; % Срединная ошибка определения температуры МЗ, град
% ошибки метеорологической подготовки
Vog.Extv = 1.6; % Срединная ошибка определения температуры воздуха, град
Vog.Ezwx = 0.87; % Срединная ошибка определения продольной составляющей
баллистического ветра на АУТ, м/с
Vog.Ezwaz = 0.87; % Срединная ошибка определения боковой составляющей
баллистического ветра на АУТ, м/с
Vog.Exwnx = 1.5; % Срединная ошибка определения продольной составляющей
баллистического ветра на ПУТ, м/с
Vog.Ezwz = 1.5; % Срединная ошибка определения боковой составляющей
баллистического ветра на ПУТ, м/с
Vog.Exhh = 2; % Срединная ошибка определения отклонения наземного давления
воздуха, мм.рт.ст.
% ошибки технической подготовки
Vog.Exnn = 1; % Срединная ошибка приспособлений для воздействия в
вертикальной плоскости, тыс.
Vog.Eznn = 0.6; % Срединная ошибка приспособлений для воздействия в
горизонтальной плоскости, тыс.
% ошибки способа расчета установок для воздействия
Vog.Ezru = 1; % Срединная ошибка способа расчета установок в горизонтальной
плоскости, тыс.

[Vog] = PodgStrel(Vog); % расчет ошибок подготовки
%-----
% задание исходных данных по способу воздействия на противодействующую систему и
порядку ведения воздействия
%-----
T = 8.0; % время выполнения задачи воздействия (мин)
%-----
nr = 300; % число реализаций (имитаций воздействия) по умолчанию
%-----
Pz = 0.30; % заданный уровень воздействия по противодействующей системе
%-----

X1min = zeros(1,length(X1)); % нижняя граница признаков наличия элемента
воздействия = 0
X1max = ones(1,length(X1)); % верхняя граница признаков наличия элемента
воздействия = 1

X2min = zeros(1,length(X2)); % нижняя граница расхода средств воздействия
каждого элемента воздействия по каждой т.в. = 0
X2max = ones(1,length(X2))*max(BK); % верхняя граница расхода средств
воздействия каждого элемента воздействия по каждой т.в. = Nmax
Xmin = cat(2,X2min,X1min);
Xmax = cat(2,X2max,X1max);
%-----
% коэффициенты матриц линейных неравенств

```

```

A1 = zeros(sum(Vk),sum(Vk));
A2 = zeros(sum(Vk),length(X2max));
for i=1:1:sum(Vk)
    for j=(i-1)*ng*nf+1:1:i*ng*nf
        A2(i,j) = 1;
    end
end
A = cat(2,A2,A1);
b = zeros(sum(Vk),1);
for i=1:1:sum(Vk)
    b(i) = 40; % максимальный комплект средств воздействия у каждого элемента
    воздействия
end

%-----
nvars = length(Xmin); % число переменных
% распределение средств воздействия по т.в.
X2 = zeros(1,length(X2min)); % инициализация начальной популяции

i = 1;
for j=1:ng*nf:length(X2)
    X2(j+i-1) = BK(i);
    i = i + 1;
    if i > ng*nf
        break
    end
end
%-----
X1 = ones(1,length(X1min)); % инициализация начальной популяции (распределение
признаков элементов воздействия)
% X = 1*ones(1,nvars); % инициализация начальной популяции
X = cat(2,X2,X1); % инициализация начальной популяции

% расчет вероятности успешности воздействия по противодействующей системе
%-----
Nsn = sum(X(1:nvars-sum(Vk))), % общий расход средств воздействия
P = -vsf(Pz,FR,GL,Fr,Gl,Kgr,Vk,ng,nf,Sn,Xc,Zc,XX,ZZ,G0,n,X,nr,nc,a,Vog), %
вероятность
%-----

N = reshape(X(1:sum(Vk)*ng*nf),[length(XX),length(ZZ),sum(Vk)]);
for k=1:1:Kgr
    N1{k} = N(:, :, sum(Vk(1:k-1))+1:sum(Vk(1:k)));
end

XX3 = X(sum(Vk)*ng*nf+1:end);

for k=1:1:Kgr
    OS{k} = XX3(sum(Vk(1:k-1))+1:sum(Vk(1:k)));
end

for k=1:1:Kgr
    for v=1:1:Vk(k)
        N1{k}(:, :, v) = OS{k}(v) .* N1{k}(:, :, v);
    end
end
%-----
% графический вывод результатов
%-----
l = 2;
for k=1:1:Kgr
    for v=1:1:Vk(k)
        figure(l); % установление текущего окна

```

```

        axes('XLim', [(-2*FR) (4*FR)], 'YLim', [(-2*GL) (4*GL)]); % установка осей
        под размеры объекта
        cla; % очистка осей
        hold on; % включение наложения
        grid on; % включение сетки
        % рисование цели (целей)
        rectangle('Position', [(PP(3)-PP(1))/2-FR/2, (PP(4)-PP(2))/2-
GL/2, FR, GL], 'Curvature', [0,0]); % границы противодействующей системы (при nc=1 -
отдельного элемента)
        if nc ~= 1
            for i=1:1:nc
                rectangle('Position', [(PP(3)-PP(1))/2+Zc(i)-Fr(i)/2, (PP(4)-
PP(2))/2+Xc(i)-Gl(i)/2, Fr(i), Gl(i)], 'Curvature', [0,0]);
            end
        end
        % рисование точек воздействия
        for i=1:1:ng
            for j=1:1:nf
                if N1{k}(i,j,v) == 0
                    plot((PP(3)-PP(1))/2+ZZ(j), (PP(4)-
PP(2))/2+XX(i), 'rs', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'g', 'MarkerSize', 3);
                else
                    plot((PP(3)-PP(1))/2+ZZ(j), (PP(4)-
PP(2))/2+XX(i), 'rs', 'MarkerEdgeColor', 'k', 'MarkerFaceColor', 'r', 'MarkerSize', 3)
                    str = num2str(N1{k}(i,j,v));
                    text((PP(3)-PP(1))/2+ZZ(j)+1, (PP(4)-PP(2))/2+XX(i), str);
                end
            end
        end
        l = l + 1;
    end
end

```

toc,

Листинг Б.2 – Функция расчета вероятности успешного воздействия по противодействующей системе методом имитационного моделирования

```

function y = vsf(Pz, FR, GL, Fr, Gl, Kgr, Vk, ng, nf, Sn, X, Z, xtp, ztp, G0, n, X2, nr, nc, a, Vog)
% расчет вероятности успешного воздействия по противодействующей системе методом
% имитационного моделирования (расчет кратных интегралов)
%-----

%-----
% формирование матриц признаков наличия элементов воздействия в группах
XX3 = X2(sum(Vk)*ng*nf+1:end);
for k=1:1:Kgr
    OS{k} = XX3(sum(Vk(1:k-1))+1:sum(Vk(1:k)));
end
%-----
% формирование матриц расхода средств воздействия для каждого элемента
воздействия по всем точкам
% воздействия
XX2 = X2(1:sum(Vk)*ng*nf);
N = reshape(XX2, [length(xtp), length(ztp), sum(Vk)]);
%-----
for k=1:1:Kgr
    N1{k} = N(:, :, sum(Vk(1:k-1))+1:sum(Vk(1:k)));
end

for k=1:1:Kgr
    for v=1:1:Vk(k)
        N1{k}(:, :, v) = OS{k}(v).*N1{k}(:, :, v);
    end
end

```

end

```
%-----
global PP;          % относительные координаты размещения осей в пикселях
                    % (относительно размеров figure)
P3 = zeros(1,nr); % выделение памяти для массива вероятностей успешности
                    % воздействия
% размеры отдельных элементов из состава противодействующей системы по фронту и
% глубине в
% пикселях
% (при nc=1 - размеры отдельного элемента)
if nc ~= 1
    for i=1:1:nc
        fr{i} = ((PP(3)-PP(1))/2+Z(i)-Fr(i)/2):1:((PP(3)-PP(1))/2+Z(i)+Fr(i)/2-
1); % фронт элемента
        gl{i} = ((PP(4)-PP(2))/2+X(i)-Gl(i)/2):1:((PP(4)-PP(2))/2+X(i)+Gl(i)/2-
1); % глубина элемента
    end
else
    fr{1} = ((PP(3)-PP(1))/2-FR/2):1:((PP(3)-PP(1))/2+FR/2-1); % фронт элемента
    gl{1} = ((PP(4)-PP(2))/2-GL/2):1:((PP(4)-PP(2))/2+GL/2-1); % глубина
    элемента
end
%-----
% координаты центра противодействующей системы
ZZZ = (PP(3)-PP(1))/2;
XXX = (PP(4)-PP(2))/2;

for ii=1:1:nr % начало цикла реализаций
    %-----
    % генерация координат случайного вектора системной ошибки подготовки
    z0 = normrnd(ZZZ,Vog.Ez(3)/0.6745);
    x0 = normrnd(XXX,Vog.Ex(3)/0.6745);
    % генерация координат случайного вектора групповой ошибки подготовки
    z1 = zeros(1,Kgr);
    x1 = zeros(1,Kgr);
    for k=1:1:Kgr
        z1(k) = normrnd(z0,Vog.Ez(2)/0.6745);
        x1(k) = normrnd(x0,Vog.Ex(2)/0.6745);
    end

    for k=1:1:Kgr
        for v=1:1:Vk(k)
            z2(k,v) = normrnd(z1(k),Vog.Ez(1)/0.6745);
            x2(k,v) = normrnd(x1(k),Vog.Ex(1)/0.6745);
        end
    end

    % задание начальных значений КЗП по противодействующей системе (отдельным
    % элементам)
    G1 = zeros(1,nc);
    if nc ~= 1
        for i=1:1:nc
            G1(i) = 0; % по каждого отдельного элемента
        end
    else
        G1(1) = 0; % по всей противодействующей системе
    end
    %-----
    % задание пустого массива условных вероятностей успешности воздействия
    Pr = [];
    %-----
    % непосредственно расчет вероятности успешности воздействия
```

```

for i2=1:1:Kgr % цикл по числу групп в системе воздействия
    for v=1:1:Vk(i2) % цикл по числу элементов воздействия в каждой группе
        if OS{i2}(v) ~= 0 %
            for j=1:1:length(xtp) % цикл по числу установок прицела
                for k=1:1:length(ztp) % цикл по числу установок угломера
                    if N1{i2}(j,k,v) ~= 0
                        for l=1:1:N1{i2}(j,k,v) % цикл по количеству средств
воздействия на каждую точку воздействия
                            % генерация координат точки падения
                            % средства воздействия
                            z3(i2,v,j,k,l) =
normrnd(z2(i2,v)+ztp(k),Vog.Vb/0.6745);
                            x3(i2,v,j,k,l) =
normrnd(x2(i2,v)+xtp(j),Vog.Vd/0.6745);
                            %-----
                            -----
                                for i1=1:1:nc % цикл по числу отдельных
элементов противодействующей системы
                                    % пересчет координат точек элемента
                                    % относительно точки падения средства
                                    % воздействия
                                    x11(i1) = (PP(4)-PP(2))/2+X(i1)-
x3(i2,v,j,k,l);
                                    z11(i1) = (PP(3)-PP(1))/2+Z(i1)-
z3(i2,v,j,k,l);
                                    % расчет КЗП в точках элемента
                                    G2(i1) =
KZPEXP(G0(i1),Sn(i1),n(i1),x11(i1),z11(i1));
                                    %-----
                                    -----
                                        Pr(i1) = 1-(1-G2(i1)).*(1-G1(i1)); %
условная вероятность успешности воздействия при фиксированных ошибках подготовки
                                        G1(i1) = Pr(i1);
                                        end
                                    end
                                end
                            end
                        end
                    end
                end
            end
        end
    end
end
%-----
% вероятность успешности воздействия по отдельным элементам
% противодействующей системы
if isempty(Pr) == 0
    if nc ~= 1
        for i=1:1:nc
            P2(ii,i) = Pr(i);
        end
        %
        % вероятность успешности воздействия по противодействующей
        % системе
        else
            P2(ii) = Pr(1);
        end
    else
        P2 = zeros(ii,nc);
    end
    %-----
end % конец цикла по числу реализаций

if nr == 1
    P = sum(a.*P2);
else
    P11 = mean(P2);

```

```

        P = sum(a.*P11);
end

y = -P;

```

```

function [G] = KZPEXP(G0,Sn,n,x,z)
% расчет координатного закона воздействия (показательная аппроксимация)
%-----
% Sn - приведенная площадь воздействия элемента
% n - параметр закона воздействия (отношение срединных отклонений)
% G0 - параметр закона воздействия средства воздействия в точке (0,0)
% x - координата x точки действия средства воздействия
% z - координата z точки действия средства воздействия
ro = 0.47694;
Ef = ro*sqrt(Sn/(pi*G0*n));
Eg = n*Ef;
G = Sn*ro^2/(pi*Eg*Ef)*exp(-ro^2*(x^2/Eg^2+z^2/Ef^2));

```

Листинг Б.3 – Функция расчета ошибок

```

function [Vog] = RassSnar(Vog)
% Функция расчета ошибок рассеивания средства воздействия и полетного времени
%-----
% Срединные ошибки рассеивания средства воздействия
% а) по дальности, м
Vog.Vd = interp1(Vog.Dc,Vog.Bd,Vog.Dcc,'spline');
% б) по направлению, м
Vog.Vb = interp1(Vog.Dc,Vog.Bb,Vog.Dcc,'spline');
% полетное время (с)
Vog.Tb = interp1(Vog.Dc,Vog.tc,Vog.Dcc,'spline');
%-----
function [Vog] = PodgStrel(Vog)
% Функция расчета ошибок подготовки воздействия
%-----
% определение угла падения
Qc = interp1(Vog.Dc,Vog.Qc,Vog.Dcc,'spline');
Qc = pi*Qc/180.0;
%-----
% Срединные ошибки вследствие ошибок определения координат и высоты
% противодействующей системы
% а) по дальности, м
if Qc ~= 0
    Eхкс = sqrt(Vog.Eхcc^2 + Vog.Ehcc^2/(tan(Qc)^2));
else
    Eхкс = Vog.Eхcc;
end
% б) по направлению, м
Ezкс = Vog.Ezcc;
%-----
% Срединные ошибки вследствие ошибок определения координат и высоты
% воздействующей позиции и ориентирования элемента воздействия
% а) по дальности, м
if Qc ~= 0
    Eхкп = sqrt(Vog.Eхpp^2 + Vog.Ehpp^2/(tan(Qc)^2));
else
    Eхкп = Vog.Eхpp;
end
% б) по направлению, м
Ezкп = sqrt(Vog.Ezpp^2 + 1e-6*Vog.Eosn^2*Vog.Dcc^2);
%-----
% Срединные ошибки вследствие ошибок баллистической подготовки

```

```

% а) по дальности, м
Exb = Vog.Exmz*0.1*interp1(Vog.Dc,Vog.dXTz,Vog.Dcc,'spline'); % ??????
проверить dXTz
% б) по направлению, м
Ezb = 0;
%-----
% Срединные ошибки вследствие ошибок метеорологической подготовки
% а) по дальности, м
dXwnx = interp1(Vog.Dc,Vog.dXwnx,Vog.Dcc,'spline');
dXhh = interp1(Vog.Dc,Vog.dXhh,Vog.Dcc,'spline');
dXtv = interp1(Vog.Dc,Vog.dXtv,Vog.Dcc,'spline');
Exmp = 0.1*sqrt(Vog.Exwnx^2*dXwnx^2+Vog.Exhh^2*dXhh^2+Vog.Extv^2*dXtv^2);
% б) по направлению, м
dZwax = interp1(Vog.Dc,Vog.dZwax,Vog.Dcc,'spline');
dZwaz = interp1(Vog.Dc,Vog.dZwaz,Vog.Dcc,'spline');
dZwnz = interp1(Vog.Dc,Vog.dZwnz,Vog.Dcc,'spline');
Ezmp =
0.0001*sqrt(Vog.Ezwax^2*dZwax^2+Vog.Ezwaz^2*dZwaz^2+Vog.Ezwnz^2*dZwnz^2)*Vog.Dcc
;
%-----
% Срединные ошибки вследствие ошибок прицельных приспособлений
% а) по дальности, м
dXtc = interp1(Vog.Dc,Vog.dXtc,Vog.Dcc,'spline');
Expp = Vog.Exnn*dXtc;
% б) по направлению, м
Ezpp = 0.001*Vog.Eznn*Vog.Dcc;
%-----
% Срединные ошибки вследствие ошибок определения установок для воздействия
% а) по дальности, м
Excr = 0.0015*Vog.Dcc;
% б) по направлению, м
Ezcr = 0.001*Vog.Ezru*Vog.Dcc;
%-----
% Срединные ошибки вследствие ошибок Таблиц воздействия
% а) по дальности, м
Extc = 0.0035*Vog.Dcc;
% б) по направлению, м
Eztc = 0;
%-----
% представление ошибок матрицей-строкой
EX = [Exkc Exkp Exb Exmp Expp Excr Extc];
EZ = [Ezkc Ezkp Ezb Ezmp Ezpp Ezcr Eztc];
%-----
% Супергрупповая ошибка подготовки воздействия
% а) по дальности, м
Vog.Ex(4) = sqrt(sum(Vog.p(1,:).*EX.^2));
% б) по направлению, м
Vog.Ez(4) = sqrt(sum(Vog.p(1,:).*EZ.^2));
%-----
% Сверхгрупповая ошибка подготовки воздействия
% а) по дальности, м
Vog.Ex(3) = sqrt(sum(Vog.p(2,:).*EX.^2));
% б) по направлению, м
Vog.Ez(3) = sqrt(sum(Vog.p(2,:).*EZ.^2));
%-----
% Групповая ошибка подготовки воздействия
% а) по дальности, м
Vog.Ex(2) = sqrt(sum(Vog.p(3,:).*EX.^2));
% б) по направлению, м
Vog.Ez(2) = sqrt(sum(Vog.p(3,:).*EZ.^2));
%-----
% Элементарная ошибка подготовки воздействия
% а) по дальности, м
Vog.Ex(1) = sqrt(sum(Vog.p(4,:).*EX.^2));

```

```
% б) по направлению, м
Vog.Ez(1) = sqrt(sum(Vog.p(4,:).*EZ.^2));
```

Листинг Б.3 – Функция ввода табличных данных по средствам воздействия

```
function [Vog] = ArtSnar(Vog)

% Функция ввода табличных данных по средствам воздействия
%-----
%
%          Массив дальностей воздействия, м
Vog.Dc = [1600 2000 3000 4000 5000 6000 7000 8000 9000 10000 11000 12000 ...
          13000 14000 15000 16000 17000 18000 19000 20000 20127];
%
%          Массив рассеиваний по дальности, м
Vog.Bd = [310 304 289 274 258 240 222 203 185 169 155 143 133 125 119 ...
          114 109 104 100 98 99];
%
%          Массив рассеиваний по направлению, м
Vog.Bb = [5.8 8.5 16 24 32 40 49 57 65 72 79 86 94 102 111 120 130 141 ...
          156 179 194];
%
%          Массив углов падения, град
Vog.Qc = [1.2 1.5 2.2 3.1 4.3 5.7 7.5 9.6 12 15 18 22 26 30 34 38 43 47 52 58
          62];
%
%          Массив изменений дальности при изменении прицела на 1
%          тыс, м
Vog.dXtc = [82 80 74 68 62 56 50 45 40 36 33 30 27 24 21 19 17 14 10 3.4 3.4];
%
%          Массив поправок прицела на продольный баллистический
%          ветер активного участка скоростью 10 м/с, тыс
Vog.dNwax = [4.0 4.2 4.9 5.7 6.7 7.8 9.1 10.6 12.1 13.6 15.1 16.7 18.5 20.5 22.5
          24.6 ...
          26.9 29.8 34.4 47.0 15.9];
%
%          Массив поправок прицела на боковой баллистический
%          ветер активного участка скоростью 10 м/с, тыс
Vog.dNwaz = [7.3 7.4 7.8 8.1 8.4 8.7 9.0 9.2 9.3 9.4 9.5 9.7 9.6 9.4 9.2 8.9 8.6
          8.8 9.3 10.8 3.3];

%=====
if Vog.dTz < 0 % отклонение ТЗ от нормальной (Тз = +15 град)
    % Массив поправок дальности на отклонения ТЗ на 10 град при отрицательном
    отклонении, м
    Vog.dXTz = [358 352 334 316 298 280 262 244 227 212 199 187 178 171 164 158
    151 143 134 118 111];
else
    % Массив поправок дальности на отклонения ТЗ на 10 град при положительном
    отклонении, м
    Vog.dXTz = [231 227 217 206 195 183 171 158 144 129 114 98 82 67 52 37 24 13
    1 -14 -20];
end
%=====

%
%          Массив поправок направления на продольный баллистический
%          ветер активного участка скоростью 10 м/с, тыс
Vog.dZwax = [1 1 1 1 1 1 2 2 2 2 2 3 3 3 4 4 5 6 7 8 9];
%
%          Массив поправок направления на боковой баллистический
%          ветер активного участка скоростью 10 м/с, тыс
Vog.dZwaz = [50 50 50 50 51 51 51 51 52 52 52 53 53 54 55 56 58 60 63 70 75];
%
%          Массив поправок дальности на продольный баллистический
%          ветер пассивного участка скоростью 10 м/с, м
Vog.dXwnx = [4 6 12 18 25 34 46 60 76 95 117 143 173 205 241 280 322 366 415 477
          496];
%
%          Массив поправок направления на боковой баллистический
%          ветер пассивного участка скоростью 10 м/с, тыс
Vog.dZwnz = [1 1 2 2 3 3 4 5 5 6 7 8 8 9 10 11 12 12 13 15 16];
%
%          Массив поправок дальности на баллистическое отклонение
%          температуры воздуха на 10 град, м
```

```

Vog.dXtv = [7 10 18 27 36 46 58 71 85 101 119 138 157 177 196 215 234 249 261
260 252];
%
%           Массив поправок дальности на отклонение наземного
%           давления воздуха на 10 мм рт ст, м
Vog.dXhh = [9 11 16 21 27 34 41 48 56 64 72 81 89 97 106 115 124 133 143 155
160];
%
%           Массив углов прицеливания, град
Vog.Alfa = [5.12 5.37 6.08 6.85 7.70 8.65 9.68 10.85 12.13 13.57 15.17 17.02
19.07 21.37 ...
23.90 26.73 29.92 33.60 38.20 45.77 50.00];
%
%           Массив окончательных скоростей, м/с
Vog.Vc = [654 638 595 554 514 474 437 403 373 348 330 318 313 310 308 310 314
318 323 333 337];
%
%           Массив полных времен полета, с
Vog.tc = [3.3 3.9 5.5 7.3 9.2 11 13 16 19 21 24 28 32 35 39 44 48 54 60 71 76];
%
%           Массив высот траектории, м
Vog.Ys = [10 15 30 52 84 132 195 278 384 519 688 903 1170 1490 1870 2340 2900
3600 4520 6150 7100];

```