

Содержание

1	Техническое задание.	2
A	Описание модели канала.	2
	Отображение примера матрицы значений a в тонах серого.	2
B	Описание работы программы.	3
C	Возможные направления развития программы.	4

1 Техническое задание.

Реализовать программу, производящую имитационное моделирование передачи сигналов в радиоканале с множеством приёмников и передатчиков. Вторичной задачей является уменьшение времени работы программы.

А Описание модели канала.

При написании программы использовалась модель со следующими параметрами:

1. Зависимость мощности сигнала от расстояния рассчитывается по следующему уравнению:

$$L = \frac{100 * P}{d^5 + a}$$

где P - Мощность передатчика.

d - Расстояние между передатчиком и приёмником.

a - Значение силы затухания в точке нахождения приёмника.

2. Значения переменной a в разных точках определяется матрицей случайных величин, коррелированных в пространстве.

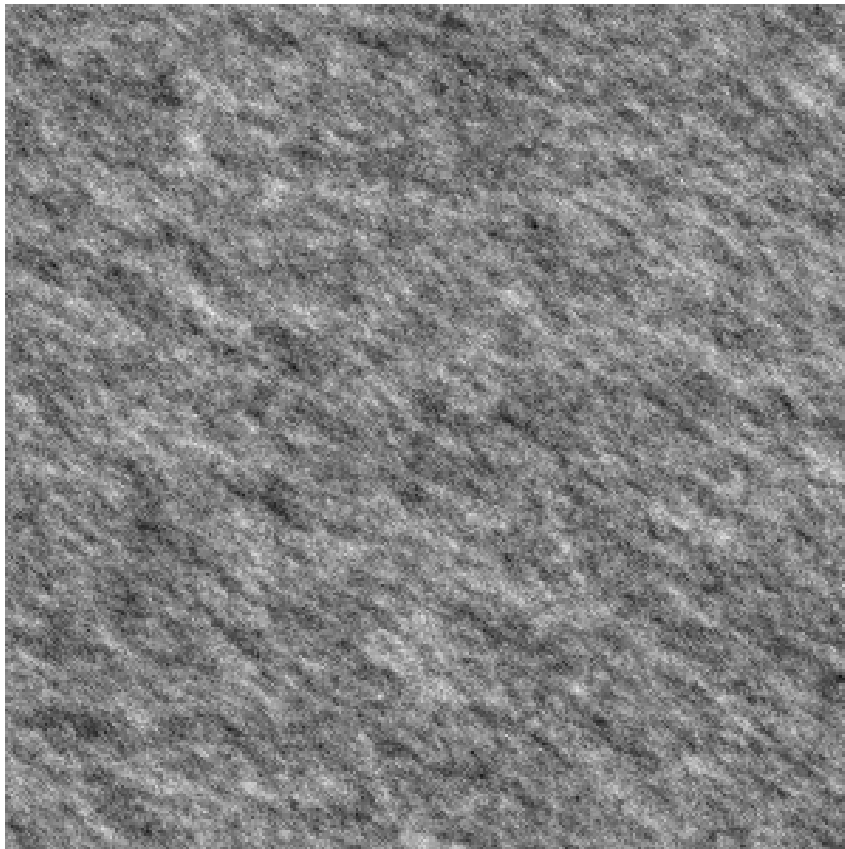


Рис. 1: Отображение примера матрицы значений a в тонах серого.

В Описание работы программы.

На вход программе подаются параметры моделирования, такие как:

1. Колчество приёмников и передатчиков.
2. Размер решётки для моделирования, высота/ширина.
3. Использование графического вывода.
4. Использование вывода в файл с указанием названия файла.
5. Ввод из файла приёмников и передатчиков.
6. Запрос краткой справки.

Генерирование координат источников и передатчиков происходит по равномерному закону распределения. На начальной стадии разработки каждому передатчику сопоставлялся приёмник (абонент). Так, при расчёте проверялось, принадлежит ли абонент данному передатчику. Если принадлежит, то мощность передатчика относилась к сигналу, если абонент не принадлежал передатчику, то мощность относилась к шуму. Результатом моделирования являются уровни отношения сигнал/шум для каждого из абонентов. При генерировании матрицы величин "а" использовалась нереалистичная модель, т.е. некоррелированные гауссовские величины. При попытке ускорения работы программы был использован метод параллельных вычислений. Была использована библиотека Posix Threads и OpenMP. При написании первой версии поточной программы не было необходимости в использовании механизмов защиты памяти от "гонок" потоков, так как запись происходила в разные сегменты памяти. Однако, ускорения работы программы не только не произошло, программа начала работать медленнее. Была рассмотрена возможная причина такого поведения программы, заключавшаяся в конфликте при считывании данных из памяти. Были так же рассмотрены несколько решений данной проблемы:

1. Выделять дополнительную память для копий конфликтных областей в отдельных потоках, передавая указатели на области памяти в потоки. Данный способ позволило выделить память с разными адресами, однако к желаемому результату это не привело.
2. Выделять память страницами, т.е., получить размер страницы памяти, выделить необходимое кол-во страниц для хранения массива и скопировать массив в данную область. Указатель на выделенную область далее передаётся в поток, к ускорению программы это также не привело.

В процессе решения данной проблемы я пытался рассмотреть такой аспект как человеческий фактор, т.е. ошибка при программировании распределения потоков. Я воспользовался библиотекой OpenMP, так как задание параметров распараллеливания в ней производится весьма простым методом (при помощи директивы компилятора `#pragma`). Но, к сожалению, данная попытка обернулась неудачей. Однако, в результате рассмотрения уже написанного кода и написания нового кода удалось добиться ускорения почти в два раза. Причина ошибки не была чётко установлена, но наиболее вероятным я считаю, что добавление директивы `inline` перед функциями, использующихся в расчёте модели, привело к такому результату. Далее была добавлена более реалистичная модель расчёта силы сигнала по следующей формуле:

$$L = 20 \cdot \log_{10}\left(\frac{4 \cdot \pi}{C \cdot f}\right) - 2 * H_r + 40 \cdot \log_{10}(d)$$

где C - скорость света (распространения сигнала в среде)

f - частота, на которой идёт передача

В программе реализован вывод на экран приемников и передатчиков. Красная антенна соответствует передатчику, жёлтая приёмнику, серая линия, соединяющая их, показывает, какой приёмник соответствует передатчику и наоборот. Рядом с приёмниками и передатчиками выводятся их порядковые номера.

С Возможные направления развития программы.

В данный момент существуют два направления развития программы:

1. Добавление функции расчёта модели на графическом процессоре для ещё более быстрого получения результатов моделирования.
2. Добавление новых возможностей в программу (переход к трёхмерной системе координат, добавление векторных шрифтов и т.д.) и исправление старых багов и ошибок в коде.