

Министерство науки и высшего образования Российской Федерации

Федеральное государственное автономное образовательное
учреждение высшего образования
Национальный исследовательский Нижегородский государственный
университет им. Н.И. Лобачевского

Институт информационных технологий, математики и механики

Отчет по лабораторной работе

«Решение систем линейных уравнений методом Гаусса-Жордана»

Выполнил:

студент группы 3824Б1ПМ1
Тончаров В.В.

Проверила:

Бусько П.В.

Нижний Новгород

Содержание

Введение.....	3
Постановка задачи.....	4
Руководство пользователя.....	5
Описание программной реализации.....	6
Результаты тестов.....	8
Заключение.....	11
Литература.....	12
Приложение.....	13

Введение

Данная лабораторная работа направлена на исследование практических аспектов реализации метода Гаусса-Жордана для решения систем линейных уравнений. Наличие различных видов математических объектов привлекает объектный метод в архитектурное решение, а наличие вариаций (решение в целых числах или в числах с плавающей запятой) мотивирует использование шаблонного программирования.

Постановка задачи

Требовалось с использованием объектного функционала C++ реализовать метод Гаусса-Жордана (с выбором ведущего элемента) для решения систем линейных уравнений.

Руководство пользователя

Программа `lineq` вызывается либо без аргументов, либо следующим образом: `lineq [опции]`, где опции – аргументы командной строки, начинающиеся с - и содержащие один или более символов `f`, `v`, `h`:

- `f` – решать в числах с плавающей точкой, а не в целых числах
- `v` – выводить в стандартный поток ошибок пошагово объясняющую алгоритм информацию, в том числе состояние матрицы на каждом шаге алгоритма
- `h` – вывести справку

Ввод программы – дополненная матрица, записанная по строкам (формат аналогичен тому, как матрица записывается на бумаге), в которой `|` выступает в качестве разделителя. Матрица может быть произвольной прямоугольной, а не только квадратной. Если уравнений больше, чем переменных, программа распознает противоречивые уравнения (с нулевой левой частью и ненулевой правой частью) в окончательной матрице. Если уравнений меньше, чем переменных, программа сочтёт столбцы, для которых не нашёлся ведущий элемент, свободными переменными. Если в дополнении матрицы несколько столбцов, матрица рассматривается как набор систем, решаемых параллельно.

Программа выводит решения уравнений в векторной форме в стандартный поток вывода, по одной системе из набора на строчку.

Описание программной реализации

Исходный код программы разделён на множество заголовочных файлов `.hxx` и исходных файлов `.cxx`. В корне находятся файл описания сборки `Makefile` и вспомогательные файлы для `clangd`.

Ошибки в программе обрабатываются главным образом при помощи исключений. Блок `catch` в функции `main` выводит ошибки удобочитаемым образом и возвращает ненулевой код завершения.

Программа состоит из следующих типов:

- `bitset` – динамический массив битов, используется в алгоритме решения для производительного запрета наличия нескольких ведущих элементов в одной строке
 - `bit_ref` – ссылка для чтения одного бита с перегруженным `operator bool`
 - `bit_ref_mut` – ссылка для чтения и записи одного бита с перегруженным `operator bool` и `operator=`
- `frac` – обыкновенная дробь, состоящая из двух 64-битных целых чисел; используется для ввода матриц и приведения строк к общему знаменателю
- `throw_only_bad` – вспомогательный класс для обработки ошибок ввода-вывода (блокирует исключения `failbit` и `eofbit`)
- `mat` – невладеющая ссылка на матрицу, обобщённая по типу элементов и порядку хранения; имеет методы для эргономичного доступа к строкам и столбцам
- `mat_maj` – порядок хранения (по строкам или по столбцам)
- `mat_mem` – владеющая ссылка на матрицу с теми же шаблонными параметрами, что и `mat`
- `mat_coord` – координаты элемента в матрице
- `mat_tpl` – вспомогательный класс для выбора правильного типа вектора в возвращаемом значении методов доступа к матрице
- `solution_traces` – «следы решения»: информация, полученная в ходе решения системы, дополняющая содержимое конечной матрицы; используется при обнаружении несовместных систем и выводе решений
- `size_if_stride` – вспомогательный класс для хранения шага между элементами вектора только в инстанциях шаблона `vec`, в которых это необходимо

- **vec** – невладеющая ссылка на строку или столбец матрицы, обобщённая по типу элементов и наличию не равного единице шага между элементами; имеет методы для эргономичного доступа к элементам и функциональной редукции

Результаты тестов

Тестирование было выполнено следующей командой:

```
for test in ../../tests/*; printf '  test %s: \n' $test;
./lineq < $test; ./lineq -f < $test; printf '\n'; end
```

Вывод:

```
test ../../tests/free_vars.txt:
x = [0 9/7 13/945 0] + t1 * [1 0 0 0] + t2 * [0 0 0 1]
x = [0 1.28571 0.0113757 0] + t1 * [1 0 0 0] + t2 * [0 0 0 1]
```

```
test ../../tests/identity.txt:
x = [1 2 3 4]
x = [1 2 3 4]
```

```
test ../../tests/redundant.txt:
```

```
equation 1: x = [1 9/2]
equation 2: x = [1 9/2]
equation 3: x = [1 9/2]
equation 4: x = [1 9/2]
equation 5: x = [1 9/2]
equation 6: x = [1 9/2]
equation 7: x = [1 9/2]
equation 8: x = [1 9/2]
equation 9: x = [1 9/2]
equation 10: x = [1 9/2]
equation 11: x = [1 9/2]
equation 12: x = [1 9/2]
equation 13: x = [1 9/2]
equation 14: x = [1 9/2]
equation 15: x = [1 9/2]
equation 16: x = [1 9/2]
equation 17: x = [1 9/2]
equation 18: x = [1 9/2]
equation 19: x = [1 9/2]
equation 20: x = [1 9/2]
equation 21: x = [1 9/2]
equation 22: x = [1 9/2]
equation 23: x = [1 9/2]
equation 24: x = [1 9/2]
equation 25: x = [1 9/2]
equation 26: x = [1 9/2]
equation 27: x = [1 9/2]
equation 28: x = [1 9/2]
equation 29: x = [1 9/2]
equation 30: x = [1 9/2]
equation 31: x = [1 9/2]
equation 32: x = [1 9/2]
equation 33: x = [1 9/2]
equation 34: x = [1 9/2]
equation 35: x = [1 9/2]
equation 36: x = [1 9/2]
equation 37: x = [1 9/2]
equation 38: x = [1 9/2]
equation 39: x = [1 9/2]
equation 40: x = [1 9/2]
equation 41: x = [1 9/2]
equation 42: x = [1 9/2]
equation 43: x = [1 9/2]
equation 44: x = [1 9/2]
equation 45: x = [1 9/2]
equation 46: x = [1 9/2]
equation 47: x = [1 9/2]
equation 48: x = [1 9/2]
equation 49: x = [1 9/2]
equation 50: x = [1 9/2]
equation 51: x = [1 9/2]
equation 52: x = [1 9/2]
equation 53: x = [1 9/2]
equation 54: x = [1 9/2]
equation 55: x = [1 9/2]
equation 56: x = [1 9/2]
equation 57: x = [1 9/2]
equation 58: x = [1 9/2]
```


equation 59: $x = [1 \ 9/2]$
 equation 60: $x = [1 \ 9/2]$
 equation 61: $x = [1 \ 9/2]$
 equation 62: $x = [1 \ 9/2]$
 equation 63: $x = [1 \ 9/2]$
 equation 64: $x = [1 \ 9/2]$
 equation 65: $x = [1 \ 9/2]$
 equation 66: $x = [1 \ 9/2]$
 equation 67: $x = [1 \ 9/2]$
 equation 68: $x = [1 \ 9/2]$
 equation 69: $x = [1 \ 9/2]$
 equation 70: $x = [1 \ 9/2]$
 equation 71: $x = [1 \ 9/2]$
 equation 72: $x = [1 \ 9/2]$
 equation 73: $x = [1 \ 9/2]$
 equation 1: $x = [1 \ 4.5]$
 equation 2: $x = [1 \ 4.5]$
 equation 3: $x = [1 \ 4.5]$
 equation 4: $x = [1 \ 4.5]$
 equation 5: $x = [1 \ 4.5]$
 equation 6: $x = [1 \ 4.5]$
 equation 7: $x = [1 \ 4.5]$
 equation 8: $x = [1 \ 4.5]$
 equation 9: $x = [1 \ 4.5]$
 equation 10: $x = [1 \ 4.5]$
 equation 11: $x = [1 \ 4.5]$
 equation 12: $x = [1 \ 4.5]$
 equation 13: $x = [1 \ 4.5]$
 equation 14: $x = [1 \ 4.5]$
 equation 15: $x = [1 \ 4.5]$
 equation 16: $x = [1 \ 4.5]$
 equation 17: $x = [1 \ 4.5]$
 equation 18: $x = [1 \ 4.5]$
 equation 19: $x = [1 \ 4.5]$
 equation 20: $x = [1 \ 4.5]$
 equation 21: $x = [1 \ 4.5]$
 equation 22: $x = [1 \ 4.5]$
 equation 23: $x = [1 \ 4.5]$
 equation 24: $x = [1 \ 4.5]$
 equation 25: $x = [1 \ 4.5]$
 equation 26: $x = [1 \ 4.5]$
 equation 27: $x = [1 \ 4.5]$
 equation 28: $x = [1 \ 4.5]$
 equation 29: $x = [1 \ 4.5]$
 equation 30: $x = [1 \ 4.5]$
 equation 31: $x = [1 \ 4.5]$
 equation 32: $x = [1 \ 4.5]$
 equation 33: $x = [1 \ 4.5]$
 equation 34: $x = [1 \ 4.5]$
 equation 35: $x = [1 \ 4.5]$
 equation 36: $x = [1 \ 4.5]$
 equation 37: $x = [1 \ 4.5]$
 equation 38: $x = [1 \ 4.5]$
 equation 39: $x = [1 \ 4.5]$
 equation 40: $x = [1 \ 4.5]$
 equation 41: $x = [1 \ 4.5]$
 equation 42: $x = [1 \ 4.5]$
 equation 43: $x = [1 \ 4.5]$
 equation 44: $x = [1 \ 4.5]$
 equation 45: $x = [1 \ 4.5]$
 equation 46: $x = [1 \ 4.5]$
 equation 47: $x = [1 \ 4.5]$
 equation 48: $x = [1 \ 4.5]$
 equation 49: $x = [1 \ 4.5]$
 equation 50: $x = [1 \ 4.5]$
 equation 51: $x = [1 \ 4.5]$
 equation 52: $x = [1 \ 4.5]$
 equation 53: $x = [1 \ 4.5]$
 equation 54: $x = [1 \ 4.5]$
 equation 55: $x = [1 \ 4.5]$
 equation 56: $x = [1 \ 4.5]$
 equation 57: $x = [1 \ 4.5]$
 equation 58: $x = [1 \ 4.5]$
 equation 59: $x = [1 \ 4.5]$
 equation 60: $x = [1 \ 4.5]$
 equation 61: $x = [1 \ 4.5]$
 equation 62: $x = [1 \ 4.5]$
 equation 63: $x = [1 \ 4.5]$
 equation 64: $x = [1 \ 4.5]$
 equation 65: $x = [1 \ 4.5]$
 equation 66: $x = [1 \ 4.5]$

```
equation 67: x = [1 4.5]
equation 68: x = [1 4.5]
equation 69: x = [1 4.5]
equation 70: x = [1 4.5]
equation 71: x = [1 4.5]
equation 72: x = [1 4.5]
equation 73: x = [1 4.5]

test ../../tests/trivial.txt:
x = 5/4
x = 1.25

test ../../tests/unsatisfiable.txt:
no solutions
no solutions
```

Все результаты соответствуют ожидаемым.

Заключение

В рамках лабораторной работы была написана удобная в пользовании программа, гибко реализующая метод Гаусса-Жордана. Объектный метод позволил сделать тело алгоритма читаемым при помощи эргономичных методов и перегруженных операций, а шаблонный функционал C++ позволил обобщить части программы и углубиться в детали решаемой задачи.

Литература

1. Керниган Б., Ритчи Д., Фьюэр А. Язык программирования СИ //М.: Финансы и статистика. – 1992.
2. Кнут Д. Э. Искусство программирования: Сортировка и поиск. – Издательский дом Вильямс, 2000. – Т. 3.

Приложение

Самый важный по мнению автора код:

```
template<typename T, mat_maj maj>
solution_traces gauss_jordan(mat<T, maj> m, size_t split, bool verbose) { // NOLINT cog. comp.
    if (split > m.n_cols()) throw range_error("matrix split out of bounds");

    solution_traces traces {m.n_rows(), m.n_cols()};
    bitset          pivots {m.n_cols()};

    for (size_t col = 0; col < split; col++) {
        size_t pivot_row = select_pivot(m.col(col), pivots);
        if (pivot_row == SIZE_MAX) {
            // No pivot found for this column - we have a free variable
            traces.free_vars()[traces.num_free_vars++] = col;
            if (verbose) cerr << "solver: column " << col << " is a free variable\n";
            continue;
        }

        if (verbose)
            cerr << "solver: selected row " << pivot_row+1 << " column " << col+1 << " as pivot\n";

        traces.pivots()[traces.num_pivots++] = {pivot_row, col};
        pivots[pivot_row] = true; // Don't use this row for pivots in the future

        for (size_t row = 0;          row<pivot_row ; row++) elim(m.row(row),m.row(pivot_row),col);
        for (size_t row=pivot_row+1; row<m.n_rows(); row++) elim(m.row(row),m.row(pivot_row),col);

        if (verbose) {
            cerr << "solver: matrix state at column " << col + 1 << ":\n";
            print_mat_with_split(cerr, m, split);
        }
    }

    for (size_t r = 0; r < m.n_rows(); r++) {
        size_t idx = m.row(r).first_nonzero_idx();
        // If there are non-zero elements in a row but all of them are to the right of the split,
        // some of the right-hand sides form unsatisfiable equations, and in those columns to
        // the right of the split where the element at a zeroed row is non-zero, the system of
        // equations is unsatisfiable.
        if (verbose) cerr << "solver: row " << r+1;
        if (idx != SIZE_MAX && idx >= split) {
            if (verbose) cerr << " has unsatisfiable equations\n";
            traces.zeroed_rows()[traces.num_0_rows++] = r;
        } else if (verbose) cerr << " doesn't have unsatisfiable equations\n";
    };

    if (verbose) {
        cerr << "solver: finished with " << traces.num_free_vars() << " free variable";
        if (traces.num_free_vars() != 1) cerr << 's';
        cerr << " and " << traces.num_0_rows() << " row";
        if (traces.num_0_rows() != 1) cerr << 's';
        cerr << " with ";
        if (m.n_cols() - split == 1) cerr << "an unsatisfiable equation\n";
        else cerr << "unsatisfiable equations\n";
    }
    return traces;
}
```