Содержание

Введение 2

1. Постановка задачи 3

2. Математические и алгоритмические основы решения задачи 5

2.1 Определитель матрицы 5

2.2 Метод Гаусса для решения систем линейных уравнений 6

2.3 Метод Гаусса для вычисления определителя 8

3. Функциональные модели и блок-схемы решения задачи 9

4. Программная реализация решения задачи 11

5. Пример выполнения программы 16

Заключение 18

Список использованных источников и литературы 19

Введение

Многие проблемы, возникающие в экономических исследованиях, планировании и управлении, будучи сформулированными математически, представляют собой задачи, в которых необходимо решить систему алгебраических уравнений.

Исторически первым, наиболее распространенным методом решения систем линейных уравнений является метод Гаусса, или метод последовательного исключения неизвестных. Сущность этого метода состоит в том, что посредством последовательных исключений неизвестных данная система превращается в ступенчатую (в частности, треугольную) систему, равносильную данной.

При практическом решении системы линейных уравнений методом Гаусса удобнее приводить к ступенчатому виду не саму систему уравнений, а расширенную матрицу этой системы, выполняя элементарные преобразования над ее строками. Последовательно получающиеся в ходе преобразования матрицы обычно соединяют знаком эквивалентности. Этот метод (который также называют методом последовательного исключения неизвестных) известен в различных вариантах уже более 2000 лет.

Помимо аналитического решения СЛАУ, метод Гаусса также применяется для нахождения матрицы, обратной к данной, определения ранга матрицы и нахождения определителя.

Целью данной курсовой работы является реализация вычисления определителя методом исключения Гаусса.

1. Постановка задачи

Пусть дана квадратная матрица A размером NxN. Требуется вычислить её определитель.

Вычисление определителя матрицы заключается в выполнении над матрицей алгоритма Гаусса для решения систем линейных алгебраических уравнений. В результате выполнения алгоритма получаем диагональную матрицу, её определитель равен произведению элементов, стоящих на диагонали.

Пример 1.

Вычислить определитель матрицы методом A исключения Гаусса.

.



Решение:

Приведем матрицу к диагональному виду методом Гаусса.

~.



Тогда определитель матрицы равен произведению ее элементов, стоящих на диагонали:

.



Знак определяется количеством обменов строк, следовательно определитель матрицы .



Пример 2.

Вычислить определитель матрицы методом A исключения Гаусса.

.



Решение:

Приведем матрицу к диагональному виду методом Гаусса.

~.



Тогда определитель матрицы равен произведению ее элементов, стоящих на диагонали:

.



Знак определяется количеством обменов строк, следовательно определитель матрицы .



2. Математические и алгоритмические основы решения задачи

2.1 Определитель матрицы

Введем определение определителя квадратной матрицы любого порядка. Это определение будет рекуррентным, то есть чтобы установить, что такое определитель матрицы порядка n, нужно уже знать, что такое определитель матрицы порядка n-1. Отметим также, что определитель существует только у квадратных матриц.

Определитель квадратной матрицы A будем обозначать или det A.



Определение. Определителем квадратной матрицы



второго порядка называется число

.



Определителем



квадратной матрицы порядка n, , называется число



где - определитель матрицы порядка n-1, полученной из матрицы A вычеркиванием первой строки и столбца с номером k.



2.2 Метод Гаусса для решения систем линейных уравнений

Пусть дана квадратная матрица A размером NxN. Требуется вычислить её определитель.

Воспользуемся идеями метода Гаусса решения систем линейных уравнений.

Дана система:

a11 x1 + a12 x2 + ... + a1n xn = b1

a21 x1 + a22 x2 + ... + a2n xn = b2

...

an1 x1 + an2 x2 + ... + ann xn = bn

Выполним следующий алгоритм.

На первом шаге найдём в первом столбце наибольший по модулю элемент, поставим уравнение с этим элементом на первую строчку (обменяв две соответствующие строки матрицы A и два соответствующих элемента вектора B), а затем будем отнимать это уравнение от всех остальных, чтобы в первом столбце все элементы (кроме первого) обратились в ноль. Например, при прибавлении ко второй строке будем домножать первую строку на -a21/a11, при добавлении к третьей - на -a31/a11, и т.д.

На втором шаге найдём во втором столбце, начиная со второго элемента, наибольший по модулю элемент, поставим уравнение с этим элементом на вторую строчку, и будем отнимать это уравнение от всех остальных (в том числе и от первого), чтобы во втором столбце все элементы (кроме второго) обратились в ноль. Понятно, что эта операция никак не изменит первый столбец - ведь от каждой строки мы будем отнимать вторую строку, домноженную на некоторый коэффициент, а во второй строке в первом столбце стоит ноль.

Т.е. на i-ом шаге найдём в i-ом столбце, начиная с i-го элемента, наибольший по модулю элемент, поставим уравнение с этим элементом на i-ю строчку, и будем отнимать это уравнение от всех остальных. Понятно, что это никак не повлияет на все предыдущие столбцы (с первого по (i-1)-ый).

В конце концов, мы приведём систему к так называемому диагональному виду:

c11 x1 = d1

c22 x2 = d2

...

cnn xn = dn

Т.е. мы нашли решение системы.

Замечание 1. На каждой итерации найдётся хотя бы один ненулевой элемент, иначе система бы имела нулевой определитель, что противоречит условию.

Замечание 2. Требование, что на каждом шаге мы выбираем наибольший по модулю элемент, очень важно в смысле численной устойчивости метода. Если выбирать произвольный ненулевой элемент, то это может привести к гигантской погрешности, когда получившееся решение будет отличаться в разы от правильного.

2.3 Метод Гаусса для вычисления определителя

Будем выполнять те же самые действия, что и при решении системы линейных уравнений, исключив только деление текущей строки на a[i][i] (точнее, само деление можно выполнять, но подразумевая, что число выносится за знак определителя). Тогда все операции, которые мы будем производить с матрицей, не будут изменять величину определителя матрицы, за исключением, быть может, знака (мы только обмениваем местами две строки, что меняет знак на противоположный, или прибавляем одну строку к другой, что не меняет величину определителя).

Но матрица, к которой мы приходим после выполнения алгоритма Гаусса, является диагональной, и определитель её равен произведению элементов, стоящих на диагонали. Знак, как уже говорилось, будет определяться количеством обменов строк (если их нечётное, то знак определителя следует изменить на противоположный). Таким образом, мы можем с помощью алгоритма Гаусса вычислять определитель матрицы за O(N3).

Осталось только заметить, что если в какой-то момент мы не найдём в текущем столбце ненулевого элемента, то алгоритм следует остановить и вернуть 0.

3. Функциональные модели и блок-схемы решения задачи

Блок-схема решения задачи представлена на рисунке 1.



Рисунок 1 – Блок-схема решения задачи для функции DETERMINATE

4 Программная реализация решения задачи

;ФУНКЦИЯ, ВЫЧИСЛЯЮЩАЯ ОПРЕДЕЛИТЕЛЬ

(DEFUN DETERMINANT (MATRIX SIZE)

;ОБЪЯВЛЕНИЕ ПЕРЕМЕННЫХ

;ОПРЕДЕЛИТЕЛЬ

(DECLARE (SPECIAL DET))

;ВСПОМОГАТЕЛЬНЫЕ МАССИВЫ И ПЕРЕМЕННЫЕ

(DECLARE (SPECIAL PAR))

(DECLARE (SPECIAL R))

(DECLARE (SPECIAL T\_))

(DECLARE (SPECIAL I))

(DECLARE (SPECIAL II))

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

(SETQ R (MAKE-ARRAY SIZE :ELEMENT-TYPE 'FLOAT :INITIAL-ELEMENT 0))

(SETQ T\_ 1)

(SETQ DET 1)

(DO

((J 0))

((>= J (- SIZE 1)))

;ИСКЛЮЧАЕМ ДЕЛЕНИЕ НА 0

(IF (= (AREF MATRIX J J) 0)

(PROGN

(SETQ II (+ J 1))

;ИЩЕМ СТРОКУ В КОТОРОЙ J-Й ЭЛЕМЕНТ НЕ 0

(DO

(())

((OR (/= (AREF MATRIX II J) 0) (= II (- SIZE 1))))

(SETQ II (+ II 1))

)

;ЕСЛИ НЕТ ТАКОЙ СТРОКИ ОПРЕДЕЛИТЕЛЬ РАВЕН 0

(IF (AND (= (AREF MATRIX II J) 0) (= II (- SIZE 1))) (SETQ T\_ 0))

;МЕНЯЕМ J СТРОКУ И НАЙДЕННУЮ

(DO

((K 0))

((>= K SIZE))

(SETF (AREF R K) (AREF MATRIX J K))

(SETF (AREF MATRIX J K) (AREF MATRIX II K))

(SETF (AREF II K) (AREF R K))

(SETQ K (+ K 1))

)

)

)

;ПРЯМОЙ ХОД

;ПРИВЕДЕНИЕ К ТРЕУГОЛЬНОМУ ТИПУ

(DO

((I (+ J 1)))

((>= I SIZE))

;ЕСЛИ (AREF MATR I J)=0 ДЕЛАТЬ НИЧЕГО НЕ НАДО

(IF (/= (AREF MATRIX I J) 0)

(PROGN

(SETQ PAR (/ (AREF MATRIX I J) (AREF MATRIX J J)))

(DO

((JJ J))

((>= JJ SIZE))

(SETF (AREF MATRIX J JJ) (\* (AREF MATRIX J JJ) PAR))

(SETF (AREF MATRIX I JJ) (- (AREF MATRIX I JJ) (AREF MATRIX J JJ)))

(SETF (AREF MATRIX J JJ) (/ (AREF MATRIX J JJ) PAR))

(SETQ JJ (+ JJ 1))

)

)

)

(SETQ I (+ I 1))

)

(SETQ J (+ J 1))

)

(IF (/= T\_ 0)

(PROGN

(DO

((I 0))

((>= I SIZE))

(SETQ DET (\* DET (AREF MATRIX I I)))

(SETQ I (+ I 1))

)

)

;ИНАЧЕ

(SETQ DET 0)

)

;ВОЗВРАЩАЕМ ОПРЕДЕЛИТЕЛЬ

DET

)

(SETQ N 0)

(SETQ INPUT\_STREAM (OPEN " D:\MATRIX.TXT" :DIRECTION :INPUT))

;РАЗМЕР МАТРИЦЫ

(SETF N (READ INPUT\_STREAM))

(SETQ MATR (MAKE-ARRAY (LIST N N) :ELEMENT-TYPE 'FLOAT :INITIAL-ELEMENT 0))

(SETF MATR (READ INPUT\_STREAM))

(CLOSE INPUT\_STREAM)

(SETQ DETERM (DETERMINANT MATR N))

;РЕЗУЛЬТАТ

(SETQ OUTPUT\_STREAM (OPEN " D:\DETERMINANT.TXT" :DIRECTION :OUTPUT))

;ЗАПИСЫВАЕМ ОПРЕДЕЛИТЕЛЬ

(PRINT 'DETERMINANT OUTPUT\_STREAM)

(PRINT DETERM OUTPUT\_STREAM)

;ЗАКРЫВАЕМ ФАЙЛ

(TERPRI OUTPUT\_STREAM)

(CLOSE OUTPUT\_STREAM)

5 Пример выполнения программы

Пример 1.

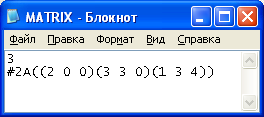


Рисунок 2 – Входные данные

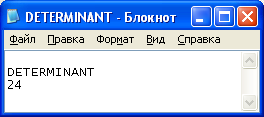


Рисунок 3 – Выходные данные

Пример 2.

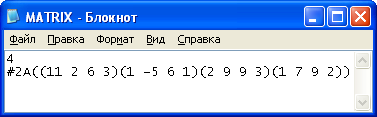


Рисунок 4 – Входные данные

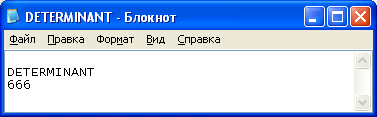


Рисунок 5 – Выходные данные

Пример 3.

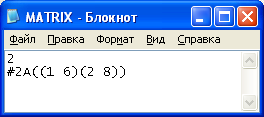


Рисунок 6 – Входные данные

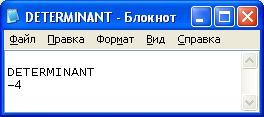


Рисунок 7 – Выходные данные

Заключение

Проблема повышения качества вычислений, как несоответствие между желаемым и действительным, существует и будет существовать в дальнейшем. Ее решению будет содействовать развитие информационных технологий, которое заключается как в совершенствовании методов организации информационных процессов, так и их реализации с помощью конкретных инструментов – сред и языков программирования.

Итогом работы можно считать созданную функциональную модель для вычисления определителя методом исключения Гаусса. Созданная функциональная модель и ее программная реализация могут служить органической частью решения более сложных задач.

Список использованных источников и литературы

1. Бронштейн, И.Н. Справочник по математике для инженеров и учащихся втузов [Текст] / И.Н. Бронштейн, К.А. Семендяев. – М.: Наука, 2007. – 708 с.
2. Васильев, Ф.П. Численные методы решения экстремальных задач. [Текст] / Ф.П. Васильев – М.: Наука, 2002. C. 415.
3. Калиткин, Н.Н. Численные методы. [Электронный ресурс] / Н.Н. Калиткин. – М.: Питер, 2001. С. 504.
4. Кнут, Д.Э. Искусство программирования. Основные алгоритмы [Текст] / Д.Э. Кнут. – М.: Вильямс, 2007. Т.1. – 712 с.
5. Метод Гаусса [Электронный ресурс] – Режим доступа: http://www.wikipedia.org/wiki/Метод\_Гаусса.
6. Степанов, П.А. Функциональное программирование на языке Lisp. [Электронный ресурс] / П.А.Степанов, А.В. Бржезовский. – М.: ГУАП, 2003. С. 79.
7. Симанков, В.С. Основы функционального программирования [Текст] / В.С. Симанков, Т.Т. Зангиев, И.В. Зайцев. – Краснодар: КубГТУ, 2002. – 160 с.
8. Хювенен Э. Мир Лиспа [Текст] / Э. Хювенен, Й. Сеппянен. – М.: Мир, 1990. – 460 с.