Лабораторная работа по курсу ”Вычислительные методы линейной алгебры”

Максимальное количество баллов: 28.

1. **Задание 1. Построение разложения Холецкого для эрмитовой матрицы. Решение СЛАУ при помощи разложения Холецкого**.

   (a) (**2 балла**) Необходимо написать программу `make_chol`, которая

   - Считывает из заданного текстового файла (вида ”Amat*.m”, где * – это какое-то число) матрицу A. Элементы матрицы A могут быть комплексными числами.
   - Проверяет, является ли матрица A квадратной, эрмитовой, с положительными ведущими минорами. Если эти условия не выполнены, то работа программы завершается с соответствующим сообщением об ошибке.
   - Строит разложение Холецкого матрицы A по формулам из лекции 8. То есть должно быть выполнено равенство $A = CC^*$, где C – нижняя треугольная матрица с положительной главной диагональю.
   - Результаты расчетов записывает в новый текстовый файл вида ”Cmat*.m”, где * – это то же число, которое было указано в названии файла с матрицей A.

   (b) (**4 балла**) Необходимо написать программу `chol_gauss`, которая

   - Считывает из заданного текстового файла (вида ”Amat*.m”, где * – это какое-то число) матрицу A.
   - Считывает из заданного текстового файла (вида ”bvec*.m”, где * – это какое-то число (то же, что и для матрицы A)) вектор b.
   - Проверяет, что количество строк матрицы A совпадает с количеством элементов вектора b. Проверяет, является ли матрица A квадратной, эрмитовой, с положительными ведущими минорами. Если эти условия не выполнены, то работа программы завершается с соответствующим сообщением об ошибке.
   - Проверяет, было ли ранее подсчитано разложение (то есть есть ли уже готовый файл ”Cmat*.m” с тем же номером *, что и у обрабатываемой матрицы A). Если подходящего файла нет, то запускается функция подсчета разложения Холецкого `make_chol`.
   - Из соответствующего файла ”Cmat*.m” считывает матрицу C.
   - При помощи считанной матрицы решает СЛАУ $Ax = b$. При этом должна быть существенно использована треугольная структура матрицы C. Не допускается решать СЛАУ $Ax = b$ при помощи алгоритма Гаусса (в общем случае) или каких-то высокоуровневых функций.
   - Результат решения – вектор x записывается в отдельный текстовый файл вида ”xvec*.m”, где * – это то же число, что было указано в матрице A.

   (c) (**2 балла**) Запустить функцию `chol_gauss` для матрицы A и вектора b очень большого размера (файлы `Amat4.m`, `bvec4.m`). Нужно 1) замерить и прислать время работы программы; 2) прислать получившийся файл `xvec4.m`.

   Замечание. Условие положительности ведущих миноров матрицы A можно не проверять предварительно. Достаточно вставить проверку на корректность расчетов по алгоритму Холецкого (чтобы не получилось деление на ноль) на каждой итерации алгоритма.

   Тестовые примеры:

   - Файлы `Amat1.m`, `Cmat1.m`. Вещественные матрицы размера 3×3. Можно использовать для тестирования функции `make_chol`.
   - Файлы `Amat2.m`, `Cmat2.m`. Комплексные матрицы размера 2×2. Можно использовать для тестирования функции `make_chol`.
   - Файлы `Amat3.m`, `bvec3.m`, `xvec3.m`. Матрица размера 3×3 и вектор длины 3. Можно использовать для тестирования функции `chol_gauss`.

   По каждой написанной программе необходимо прислать на проверку и продемонстрировать исходный код. Кроме того, в пункте (c) нужно прислать получившийся файл `xvec4.m` и (в отдельном файле ’task_1c.txt’) указать время работы программы на данном тесте.

2. **Задание 2. Построение QR-разложения матрицы. Решение СЛАУ при помощи QR-разложения матрицы**.

   (a) (**6 баллов**) Необходимо написать программу `make_qr`, которая

   - Считывает из заданного текстового файла (вида ”Amat*.m”, где * – это какое-то число) номер используемого численного метода (в первой строке: `"(Method=*)."`), а также матрицу A с вещественными элементами.

- Проверяет, является ли матрица A квадратной (если это условие не выполнено, то работа программы завершается с сообщением об ошибке).
- Строит QR-разложение матрицы по формулам из лекции 9 одним из 3 способов (в программе должны быть реализованы все три способа!):
  - Метод отражений (`Method=1`).
  - Метод вращений (`Method=2`).
  - Алгоритм Грама-Шмидта (`Method=3`).
- Результаты расчетов записывает в двух новых текстовых файлах вида "Qmat*.m" и "Rmat*.m", где * – это то же число, которое было указано в названии файла с матрицей A.

(b) (**4 балла**) Необходимо написать программу `qr_gauss`, которая

- Считывает из заданного текстового файла (вида "Amat*.m", где * – это какое-то число) номер используемого численного метода (в первой строке: "`(Method=*).`"), а также матрицу A с вещественными элементами.
- Считывает из заданного текстового файла (вида "bvec*.m", где * – это какое-то число (то же, что и для матрицы A)) вектор b с комплексными или вещественными элементами.
- Проверяется, что число строк матрицы A совпадает с числом элементов вектора b, что матрица A является квадратной и невырожденной.
- Проверяет, было ли ранее (любым методом) подсчитано QR-разложение (есть ли уже готовые файлы "Qmat*.m", "Rmat*.m"). Если подходящих файлов нет, то запускается функция подсчета QR-разложения (программа `make_qr`).
- Из соответствующего файла считываются матрицы Q, R.
- При помощи считанных матриц решается СЛАУ $Ax = b$. При этом должна быть существенно использована унитарность матрицы Q, а также треугольная структура матрицы R. Не допускается решать СЛАУ $Ax = b$ при помощи алгоритма Гаусса (в общем случае).
- Результат решения – вектор x записывается в отдельный текстовый файл вида "xvec*.m", где * – это то же число, что было указано в матрице A.

(c) (**2 балла**) Запустить функцию `qr_gauss` для матрицы A и вектора b очень большого размера (файлы `Amat9.m`, `bvec9.m`). Нужно 1) замерить и прислать время работы программы; 2) прислать получившийся файл `xvec9.m`.

Тестовые примеры:

- Файлы `Amat5.m`, `Qmat5.m`, `Rmat5.m`. Матрицы размера 3×3, метод 1. Можно использовать для тестирования функции `make_qr`.
- Файлы `Amat6.m`, `Qmat6.m`, `Rmat6.m`. Матрицы размера 3×3, метод 2. Можно использовать для тестирования функции `make_qr`.
- Файлы `Amat7.m`, `Qmat7.m`, `Rmat7.m`. Матрицы размера 3×3, метод 3. Можно использовать для тестирования функции `make_qr`.
- Файлы `Amat8.m`, `bvec8.m`, `xvec8.m`. Матрица размера 3×3 и вектор длины 3. Можно использовать для тестирования функции `qr_gauss`.

По каждой написанной программе необходимо прислать на проверку и продемонстрировать исходный код. Кроме того, в пункте (c) нужно прислать получившийся файл `xvec9.m` и (в отдельном файле 'task_2c.txt') указать время работы программы на данном тесте.

3. **Задание 3. Поиск собственных значений матрицы. Построение характеристического многочлена матрицы**.

(a) (**8 баллов**) Необходимо написать программу `find_poly_eig`, которая

- Считывает из заданного текстового файла (вида "Amat*.m", где * – это какое-то число) матрицу A.
- Проверяет, является ли матрица A квадратной (если это условие не выполнено, то работа программы завершается с сообщением об ошибке).
- Проверяет, является ли матрица A эрмитовой.
- Если матрица A является эрмитовой, то при помощи итерационного метода вращений (лекция 12) программа приближённо подсчитывает вектор из собственных значений матрицы A. Результаты расчетов программа записывает в новом текстовом файле вида "cvec*.m", где * – это то же число, которое было указано в названии файла с матрицей A.
- Если матрица A не является эрмитовой, то к ней применяется метод элементарных преобразований (лекция 13). В результате должна быть получена либо трёхдиагональная матрица (если возможно), либо верхняя почти треугольная матрица B. Найденная матрица записывается в виде файла

"Bmat*.m", где * – это то же число, которое было указано в названии файла с матрицей A. Если удалось подсчитать трёхдиагональную матрицу, то для неё также рассчитываются коэффициенты характеристического многочлена, которые в виде вектора записываются в новый файл вида "Acoeff*.m", где * – это то же число, которое было указано в названии файла с матрицей A.

Замечание. Предполагается, что характеристический многочлен должен быть представлен в нормированном виде

$$f(\lambda) = \lambda^n + a_1\lambda^{n-1} + a_2\lambda^{n-2} + \ldots + a_{n-1}\lambda + a_n,$$

а в файл "Acoeff*.m" должны быть записаны коэффициенты $[a_1; a_2; \ldots; a_n]$ (то есть без учёта первого, единичного коэффициента).

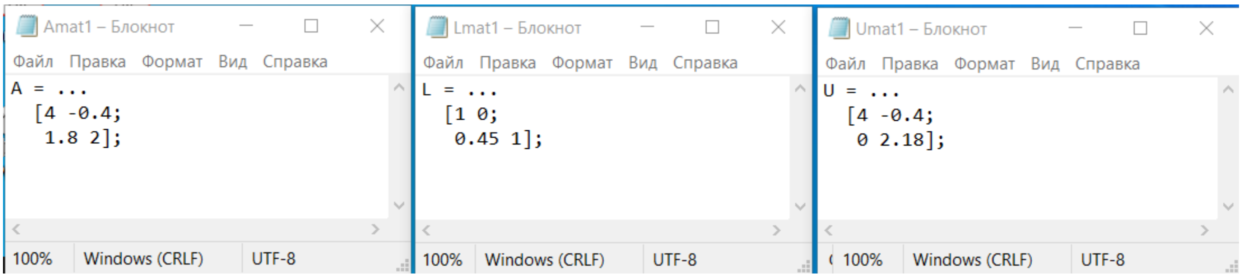
Тестовые примеры:

- Файлы `Amat10.m`, `cvec10.m`. Эрмитова матрица размера 3×3 и вектор из собственных значений.
- Файлы `Amat11.m`, `Acoeff11.m`. Неэрмитова матрица размера 3×3 и вектор из коэффициентов характеристического многочлена.

Необходимо прислать на проверку и продемонстрировать исходный код программы.

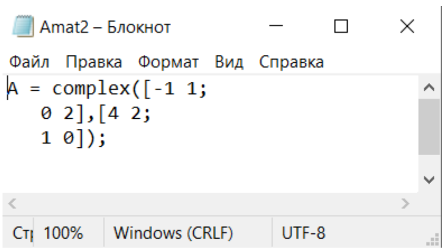
Общие замечания:

- Перед тем, как начинать программировать конкретные алгоритмы рекомендуется написать вспомогательные функции для чтения и записи в текстовые файлы матриц. Эти функции (одни и те же) далее могут быть использованы в каждом из заданий. Каждая матрица записывается в отдельном текстовом файле, например, вида "Amat1.m". Пример:



Здесь представлены три файла с матрицами размера 2×2. Первая строка файла всегда содержит имя матрицы, знак равенства и три точки. Конструкция с числами матрицы заключается внутрь квадратных скобок. Каждая строка матрицы задаётся отдельно, в одной строке файла. Между числами строки указываются символы-разделители 'пробел'. Символ ';' указывает на конец строки (переход к следующей строке). Этот же символ указывается в самом конце матрицы, после квадратной скобки. При записи действительных чисел в качестве символа, отделяющего целую часть числа от дробной, используется 'точка' ('.').

Матрицы и векторы с комплексными значениями содержат по две отдельные компоненты, соответствующие вещественной и мнимой частям чисел:



В этом примере в файле записана матрица

$$A = \begin{pmatrix} -1+4i & 1+2i \\ i & 2 \end{pmatrix}.$$

- В программах **запрещено использовать готовые функции**, позволяющие строить разные разложения матриц, обращать матрицы, искать определители или ранги матриц и т.п. То есть все вычисления должны быть произведены только за счет использования элементарных операций.

- В программах следует воздержаться от "плохих" операций, приводящих к большим погрешностям вычислений (например, от деления на очень малые числа). Уровень "очень малого числа" можно задать самостоятельно в качестве параметра. Также следует помнить, что при работе с вещественными числами операции сравнения чисел на равенство являются бессмысленными, а потому их надо заменять на более осмысленные сравнения типа неравенство.