Отчет о выполненнном задании по численным методам линейной алгебры.

Егор Подлесов

15 декабря 2023 г.

Краткий обзор.

Требуется:

1. Реализовать метод для оценки спектра матрицы и итерационный метод Чебышева.

2. Оценить спектр матрицы.

3. Определить количество итераций в итерационном методе Чебышева так, чтобы погрешность была меньше чем у прямого метода вращений Гивенса.

4. Среднеквадратическую норму погрешности решения прямого метода и метода Чебышева, относительную погрешность решения метода Чебышева.

5. График среднеквадратической нормы погрешности решения.

Реализация.

```
template <typename T>
TVector<T> GetParameters(const TMatrix<T>& a, size_t
   iter_num) {
    TVector<T> result(iter_num);
    result(0) = 1;
    for (size_t i = 1; i < iter_num; i *= 2) {
        TVector<T> tmp(iter_num);
        for (size_t j = 0; j < 2 * i; ++j) {
            if (j & 1) {
                tmp(j) = (i << 2) - result((j - 1) >> 1);
            } else {
                tmp(j) = result(j >> 1);
            }
        }
        result = std::move(tmp);
    }
    const auto [spectrum_min, spectrum_max] = a.
       GetSpectrumBoundaries();
    const auto r_0 = (spectrum_max - spectrum_min) / (
       spectrum_max + spectrum_min);
    const auto tau_0 = 2 / (spectrum_min + spectrum_max);
    for (size_t i = 0; i < iter_num; ++i) {
```

```
        result(i) = -std::cos(result(i) / (iter_num << 1)
            * M_PI);
        result(i) = tau_0 / (1 + result(i) * r_0);
    }
    return result;
}

template <typename T>
bool SolveSystem(const TMatrix<T>& a, const TVector<T>& b
   , TVector<T>& result, const size_t iter_num) {
    if (a.GetSize1() != a.GetSize2()) {
        return false;
    }
    auto methodParams = GetParameters(a, iter_num);
    result = TVector<T>(b.GetSize());
    result.Nullify();
    for (size_t iter = 0; iter < iter_num; ++iter) {
        TVector<T> tmp(result.GetSize());
        tmp.Nullify();
        for (size_t i = 0; i < a.GetSize1(); ++i) {
            double v = 0;
            for (size_t j = 0; j < a.GetSize2(); ++j) {
                v += a(i, j) * result(j);
            }
            tmp(i) = (b(i) - v) * methodParams(iter_num)
               + result(i);
        }
        result = std::move(tmp);
    }
    return true;
}

template <typename T>
std::pair<T, T> TMatrix<T>::GetSpectrumBoundaries() const
    {
    double left = 0;
    double right = 0;
    for (size_t i = 0; i < Size1; ++i) {
        double r = 0;
        for (size_t j = 0; j < Size2; ++j) {
            if (j == i) {
                continue;
            }
```

```
            r += fabs(Data[i * Size2 + j]);
        }
        left = std::min<T>(left, Data[i * Size2 + i] - r)
            ;
        right = std::max<T>(right, Data[i * Size2 + i] +
            r);
    }
    return std::make_pair(left, right);
}
```

Оценка спектра матрицы.

Спектр матрицы будем оценивать по теореме Гершгорина.

Положим

$$R_i = \sum_{i \neq j} |a_{ij}|$$

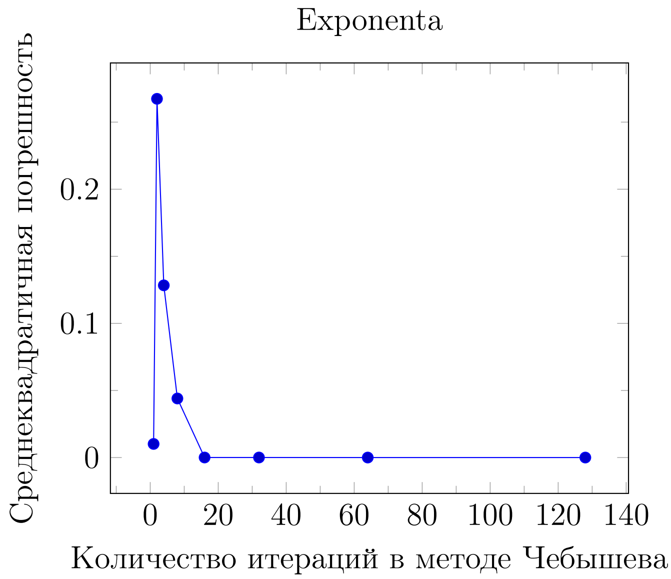
$D(a_{ii}, R_i)$ - круг с центром в точке a_{ii}(для комплексного случая) и радиусомR_i

Теорема Гершгорина

Каждое собственное значение матрицы лежит хотя бы в одном круге Гершгорина.

Сравнение прямого метода вращений и итерационного метода Чебышева.

Метод решения	Среднеквадратичная погрешность	Относительная погрешность
Метод Гаусса	0.00359501	0.00792001
Метод Чебышева, 1	0.010141	0.0311231
Метод Чебышева, 2	0.267352	0.801734
Метод Чебышева, 4	0.128336	0.398931
Метод Чебышева, 8	0.04401372	0.129781
Метод Чебышева, 16	0.00301129	0.00917512
Метод Чебышева, 32	3.19325e-05	5.67183e-05
Метод Чебышева, 64	7.74915e-10	2.974971e-09
Метод Чебышева, 128	9.01374e-19	3.83482e-18



Из вышеописанного видно, что итерационный метод выигрывает у прямого метода решения СЛАУ (метода вращений) при количестве итераций $iter\_num \geq 32$

Вывод.

Реализован метод итераций Чебышева. Оценена его сходимость. Проведено сравнение с прямым методом вращений.