Полная документация к библиотеке по работе с матрицами. Оформлена при помощи LaTeX

Студент: Карасева Ульяна
Группа: 151

1 Вступление.

Библиотека для работы с матрицей — полезная и широко используемая вещь, упрощающая вычислительные действия и сокращающая время, потраченное на эти вычисления. Все представленные функции просты, но объемны при больших размерах матриц. Именно поэтому представленная библиотека так необходима.

- Данная библиотека создавалась с целью ее дальнейшего использованию для работы с матрицами.

- Основные объекты, фигурирующие в использовании данной библиотеке — матрицы, составленные из вещественных чисел.

- Библиотека содержит несколько функций, необходимых в линейной алгебре для работы с матрицами.

1.1 Теория.

Матрица — математический объект, записываемый в виде прямоугольной таблицы элементов кольца или поля (например, целых, действительных или комплексных чисел), которая представляет собой совокупность строк и столбцов, на пересечении которых находятся её элементы. Количество строк и столбцов задает размер матрицы.
Для матрицы определены следующие алгебраические операции:

- сложение матриц, имеющих один и тот же размер;
- умножение матриц подходящего размера (матрицу, имеющую n столбцов, можно умножить справа на матрицу, имеющую n строк);
- в том числе умножение на матрицу вектора (по обычному правилу матричного умножения; вектор является в этом смысле частным случаем матрицы);
- умножение матрицы на элемент основного кольца или поля (то есть скаляр).

Минором M_{ij} к элементу a_{ij} определителя n-го порядка называется определитель $(n-1)$-го порядка, полученный из исходного вычеркиванием i-той строки и j-того столбца.

Алгебраическим дополнением A_{ij} к элементу a_{ij} определителя n-го порядка называется число $A_{ij} = (-1)^{i+j} \cdot M_{ij}$
Транспонированная матрица — матрица A^T, полученная из исходной матрицы A заменой строк на столбцы.
Для квадратной матрицы $A = (a_{ij})$ размера $n \times n$ её **определитель** $detA$ вычисляется по формуле:

$$detA = \sum_{\alpha_1, \alpha_2, \ldots, \alpha_n} (-1)^{N(\alpha_1, \alpha_2, \ldots, \alpha_n)} \cdot a_{1_{\alpha_1}} a_{2_{\alpha_2}} \ldots a_{n_{\alpha_n}}$$

Обратная матрица — такая матрица A^{-1}, при умножении на которую исходная матрица A даёт в результате единичную матрицу E.
Метод Гаусса — классический метод решения системы линейных алгебраических уравнений.

Описание алгоритма

Алгоритм решения СЛАУ методом Гаусса подразделяется на два этапа.

- На первом этапе осуществляется так называемый прямой ход, когда путём элементарных преобразований над строками систему приводят к ступенчатой или треугольной форме, либо устанавливают, что система несовместна. А именно, среди элементов первого столбца матрицы выбирают ненулевой, перемещают его на крайнее верхнее положение перестановкой строк и вычитают получившуюся после перестановки первую строку из остальных строк, домножив её на величину, равную отношению первого элемента каждой из этих строк к первому элементу первой строки, обнуляя тем самым столбец под ним. После того, как указанные преобразования были совершены, первую строку и первый столбец мысленно вычёркивают и продолжают пока не останется матрица нулевого размера. Если на какой-то из итераций среди элементов первого столбца не нашёлся ненулевой, то переходят к следующему столбцу и проделывают аналогичную операцию.

- На втором этапе осуществляется так называемый обратный ход, суть которого заключается в том, чтобы выразить все получившиеся базисные переменные через небазисные и построить фундаментальную систему решений, либо, если все переменные являются базисными, то выразить в численном виде единственное решение системы линейных уравнений. Эта процедура начинается с последнего уравнения, из которого выражают соответствующую базисную переменную (а она там всего одна) и подставляют в предыдущие уравнения, и так далее, поднимаясь по «ступенькам» наверх. Каждой строчке соответствует ровно одна базисная переменная, поэтому на каждом шаге, кроме последнего (самого верхнего), ситуация в точности повторяет случай последней строки.

2 Функции и методы их использования.

Библиотека содержит следующие функции:

- **GetMatr** — функция, считающая миноры матрицы.

```
  Matr<T> GetMatr(int i_d, int j_d)
{
  Matr<T> matr = *this;
  int i, j, i1 = 0, j1 = 0;

  matr.size.n = matr.size.n - 1;
  matr.size.m = matr.size.m - 1;

  matr.m.erase(matr.m.begin() + i_d);

  for (i = 0; i < size.n - 1; i++)
    matr.m[i].erase(matr.m[i].begin() + j_d);

  return matr;
}
```

- **s_alg** — функция, считающая алгебраические дополнения.

```
  Matr<T> s_alg(int x, int y)
{
  int i, j;
  Matr<T> m_new(size);

  for (i = 0; i < size.n; i++)
    for (j = 0; j < size.n; j++)
m_new.m[i][j] = i == y && j == x ? 1 : i == y || j == x ? 0 : m[i][j];

  return m_new;
}
```

- **Transpose** — функция, возвращающая транспонированную матрицу.

```
  Matr<T> Transpose()
{
  Matr<T> t(Size(size.m, size.n));

  for (size_t i = 0; i < size.n; i++)
    for (size_t j = 0; j < size.m; j++)
      t.m[j][i] = m[i][j];
  return t;
}
```

- **Determinant** — функция, считающая определитель матрицы.

```
  T Determinant() {
  int i, j, d, k;
  int n;
  Matr<T> p(size);
  assert(size.n == size.m);
  j = 0; d = 0;
  k = 1;
  n = size.m - 1;

  if (size.m == 1) {
    d = m[0][0];
    return(d);
  }
  if (size.m == 2) {
    d = m[0][0] * m[1][1] - m[1][0] * m[0][1];

    return(d);
  }
  if (size.m>2) {
    for (i = 0; i<size.m; i++) {
      p = GetMatr(i, 0);
      d = d + k * m[i][0] * p.Determinant();
      k = -k;
    }
  }
  return(d);
}
```

- **Inverse** — функция, считающая обратную матрицу.

```
    Matr<T> Inverse()
  {
    int i, j;
    Matr<T> m_t = *this;
    double k = m_t.Determinant();
    if(k==0)
        cout<<"Can't find inverse"<<endl;
    for (i = 0; i < size.n; i++)
      for (j = 0; j < size.n; j++)
      {
        /* Building algebraic transpose adjunct */
        Matr<T> tmp = s_alg(i, j);

        m_t.m[i][j] = tmp.Determinant();
      }

    for (i = 0; i < size.n; i++)
      for (j = 0; j < size.n; j++)
        {
            m_t.m[i][j]=m_t.m[i][j]/k;
        }

    return m_t;
  }
```

- **gauss** — функция, выполняющая метод Гаусса для решения СЛУ.

```
    Matr(){}
double* gauss(T *y,ostream &out)
{
  T max;
  int k, index;
  const double eps = 0.00001;
  k = 0;
  while (k < size.n)
  {

    max = abs(m[k][k]);
    index = k;
    for (int i = k + 1; i < size.n; i++)
    {
      if (abs(m[i][k]) > max)
      {
        max = abs(m[i][k]);
        index = i;
      }
    }

    if (max < eps)
    {

      cout << "There is no solution because of";
      cout << index << " column of the matrix" << endl;

    }
    for (int j = 0; j < size.n; j++)
    {
      double temp = m[k][j];
      m[k][j] = m[index][j];
      m[index][j] = temp;
    }
    double temp = y[k];
    y[k] = y[index];
    y[index] = temp;

    for (int i = k; i < size.n; i++)
    {
      double temp = m[i][k];
      if ((temp < eps)& (temp > -eps)) continue;
      for (int j = 0; j < size.n; j++)
        m[i][j] = m[i][j] / temp;
      y[i] = y[i] / temp;
      if (i == k)  continue;
      for (int j = 0; j < size.n; j++)
        m[i][j] = m[i][j] - m[k][j];
```

```
        y[i] = y[i] - y[k];
      }
      k++;
    }

    for (k = size.n - 1; k >= 0; k--)
    {
      x[k] = y[k];
      for (int i = 0; i < k; i++)
        y[i] = y[i] - m[i][k] * x[k];

    }
    return x;
  }
  T &elem(int i, int j) {return m[i][j];}
};
```

- **Print** — функция, выводящая матрицу на экран.

```
    void Print(ostream &out)
  {
    out << "matr is:" << endl;
    for (size_t i = 0; i < size.n; i++)
      for (size_t j = 0; j < size.m; j++)
        if (j == size.m - 1)
          out << m[i][j] << endl;
        else
          out << m[i][j] << ' ';
```

Также в библиотеке реализованы основные алгебраические операции над матрицами:

- сложение/разность матриц, имеющих один и тот же размер;
- умножение двух матриц подходящего размера;
- умножение матриц на число.

3 Основные принципы работы

- Для получения ответа на поставленную задачу пользователю необходимо:

    1. Ввести матрицы в текстовый файл, данные которых считывает библиотека, и выбрать, что он хочет решить.
    2. Далее следует выбрать функцию из библиотеки, соответствующую нужной задаче.
    3. Через функцию Print вывести полученный результат в файл.

- В консоль будет выводиться информация об ошибках, из-за которых невозможно воспользоваться той или иной функцией.

4 Классификация возможных ошибок.

5 Использованные при создании библиотеки источники и материалы.

- https://ru.wikipedia.org/wiki/