Нижегородский государственный университет им. Н. И. Лобачевского

Радиофизический факультет
Кафедра математики

Отчет по лабораторной работе:

Решение систем линейных алгебраических уравнений

Введение в численные методы

Выполнил: студент 421 группы **Зайцев Юрий**

Проверил: **Кулинич Виктор Валентинович**

Нижний Новгород
2006 год

Содержание

Введение

В настоящей работе анализируется процесс решения заданной системы линейных уравнений с помощью численных методов на персональном компьютере. Для этого используются два стандартных метода решения линейных алгебраических уравнений: *прямой* метод исключения Гаусса и *итерационный* метод Гаусса-Зейделя. В ходе решения будут учитываться неустранимые априорные погрешности исходной системы, которые, как правило, имеют место в реальных физических экспериментах. В конце работы будет произведён общий анализ полученных результатов.

1 Постановка учебно-практической задачи

Целью данной лабораторной работы являлось решение системы линейных уравнений

$$Ax = b, \tag{1}$$

заданной формулой

$$x_i = \frac{1}{17}\sum_{j=1}^{n} x_j \sin j + \cos i\alpha, \quad \alpha \in \left[-\frac{\pi}{2}; \frac{\pi}{2}\right], \quad i = \overline{1, n}. \tag{2}$$

При этом, в качестве невозмущенного состояния системы было выбрано значение $\alpha = -\dfrac{\pi}{100} \simeq -0.0314$, заданное с относительной погрешностью $\delta\alpha = 5\%$. Определявшая точность решения системы константа ε была выбрана равной $\varepsilon = 10^{-5}$. Значения n (числа неизвестных), для которых необходимо было решить систему определялись набором $n \in \{5, 10, 15\}$.

Для решения системы методом Гаусса необходимо было использовать программу gaudemo.for из каталога LIBR, либо воспользоваться стандартной программой из пакета Matlab. Для написания программы для решения системы методом Гаусса-Зейделя либо воспользоваться пакетом Matlab, либо любой другой средой программирования. Результаты решения системы вместе с оценками ошибок представить в графической форме, кроме того, для каждого случая оценить число обусловленности матрицы, проверить критерии сходимости метода и вычислить оптимальное число итераций для обеспечения необходимой точности решения.

2 Описание алгоритма

Итак, систему (2) требуется решить прямым методом Гаусса и итерационным методом Гаусса-Зейделя, и сравнить полученные решения.

Для большей наглядности, приведем запись системы (2) в матричном виде для системы, содержащей, например, три неизвестных ($n = 3$):

$$\begin{pmatrix} 17-\sin 1 & -\sin 2 & -\sin 3 \\ -\sin 1 & 17-\sin 2 & -\sin 3 \\ -\sin 1 & -\sin 2 & 17-\sin 3 \end{pmatrix} x = 17 \begin{pmatrix} \cos\alpha \\ \cos 2\alpha \\ \cos 3\alpha \end{pmatrix} \tag{3}$$

Несложно заметить, что расширенную матрицу такой системы можно построить при помощи следующего фрагмента кода Matlab, который, собственно, и использовался в программе:

```
for i = 1:1:n
    a(i, i) = 1;
    for j = 1:1:n
        a(i, j) = a(i, j) - sin(j) / 17;
    end
    b(i) = cos(alpha * i);
end
```

2.1 Прямой метод Гаусса

Метод Гаусса заключается в приведении системы к системе уравнений с треугольной матрицей (прямой ход метода) и последующим решением этой системы, начиная с последнего уравнения (обратный ход метода). Прямой метод Гаусса реализован с помощью встроенной функции нахождения обратной матрицы математического пакета Matlab.

2.2 Итерационный метод Гаусса-Зейделя

Решение системы (2) итерационным методом получается как предел последовательности векторов $\{x^k\}_{k=1}^{\infty}$, каждый из которых вычисляется по единому правилу. Начальный элемент x^1 выбирается в принципе произвольно, однако имеет смысл задать его по порядку величины близким к ожидаемому результату, ведь это может существенно уменьшить число итераций, необходимых для нахождения решения с требуемой точностью. В данной работе x^1 был выбран равным $x^1 = b$.

Для итерационного метода, как правило, существует такая последовательность невырожденных матриц H_k, $k = 1, 2, \ldots$, что правило построения элементов итерационной последовательности записывается в виде

$$x^{k+1} = T_k x^k + H_k b, \tag{4}$$

где $T_k = E - H_k A$, E — единичная матрица.

Используемый метод Гаусса-Зейделя является стационарным, т.е. матрица $H_k = H$ не зависит от номера шага k. Для данного метода

$$H_k = (D + L)^{-1}, \quad T_k = -(D + L)^{-1} R,$$

где $L + D + R = A$, а L – нижняя треугольная матрица с нулевыми диагональными элементами, D – диагональная матрица, R – верхняя треугольная матрица с нулевыми диагональными элементами. Таким образом, итерационный процесс имеет вид

$$x^{k+1} = -(D + L)^{-1} R x^k + (D + L)^{-1} b. \tag{5}$$

Однако формулу (5) можно записать в покоординатной форме и программировалась, в нашем случае, именно она:

$$\begin{array}{l} x_1^{k+1} = (b_1 - a_{12}x_2^k - \cdots - a_{1n}x_n^k)/a_{11} \\ x_2^{k+1} = (b_2 - a_{21}x_1^{k+1} - \cdots - a_{2n}x_n^k)/a_{22} \\ \\ \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots \\ \\ x_n^{k+1} = (b_n - a_{n1}x_1^{k+1} - \cdots - a_{nn-1}x_{n-1}^{k+1})/a_{nn} \end{array} \tag{6}$$

С помощью итерационного метода ищется приближенное решение системы. Говорят, что итерация x^k является с точностью ε приближенным решением системы, если

$$\|x^k - x^0\| \le \varepsilon,$$

где x^0 — точное решение системы. Под нормой $\|x\|$ вектора $x = \{x_1, ..., x_n\} \in R^n$ здесь понимается $\|x\| = \|x\|_2 = \left(\sum\limits_{i=1}^{n} |x_i|^2\right)^{1/2}$.

Очевидно, что для того, чтобы стационарный итерационный процесс (4) сходился, достаточно, чтобы выполнялось неравенство

$$\|T\| < 1. \tag{7}$$

Здесь под нормой $(n \times n)$ – матрицы A понимается $\|A\| = \|A\|_* = \left(\sum\limits_{i,j=1}^{n} |a_{ij}|^2\right)^{1/2}$, то есть эта норма является согласованной с нормой $\|x\|_2$, введенной ранее (для любых x и A выполняется $\|Ax\|_2 \le \|A\|_* \|x\|_2$). Кроме того, эта норма является подчиненной норме $\|x\|_2$ (то есть для любой матрицы A выполняется $\|A\| = \sup\limits_{x \ne 0} \dfrac{\|Ax\|_2}{\|x\|_2}$).

Таким образом, критерием окончания итерационного процесса при достижении заданной точности может являться неравенство

$$\|x^{k+1} - x^k\| \le \varepsilon \frac{1 - \|T_k\|}{\|T_k\|}. \tag{8}$$

3 Исследование применимости алгоритма

Как упоминалось выше (в описании алгоритма), для сходимости итерационного процесса достаточно выполнение условия (7). Кроме того, известно, что для существования единственного решения системы (1) и сходимости метода Гаусса-Зейделя достаточно выполнения условия

$$\sum_{i\neq j} |a_{ij}| < |a_{ii}|\,, \;\; i = 1,2,...,n. \tag{9}$$

Проверим выполнение этого условия для заданной системы (2):

$$\sum_{i\neq j} |a_{ij}| = \sum_{i\neq j} |-\sin j| \leq n-1 \leq 14, \text{ при } n = 1\dots 15.$$

С другой стороны,

$$|a_{ii}| = |17-\sin i| \quad \Rightarrow \quad 16 \leq |a_{ii}| \leq 18$$

Таким образом, очевидно, что при заданных $n = \{5, 10, 15\}$ условие (9) выполняется:

$$\sum_{i\neq j} |a_{ij}| \leq 14 < 16 \leq |a_{ii}| \leq 18, \quad i = 1,2,...,n.$$

Для нашей системы (2) при $n = \{5, 10, 15\}$ норма матрицы T принимала следующие значения: $\|T\| = \{0.1506, 0.2611, 0.4254\} < 1$. Поэтому наш итерационный процесс (4) сходится, и, в качестве критерия его окончания, мы можем использовать неравенство (8).

3.1 Обусловленность линейной системы

В работе также рассматривается случай неточных входных данных, соответствующий задаче решения линейной системы, составленной из полученных в результате физического эксперимента параметров с априорно определяемой погрешностью. Входным параметром в данном случае является норма возмущения вектора свободных членов $\|\Delta b\|$, а в ходе работы осуществляется оценка нормы максимального вектора отклонения точного решения $\|\Delta x\|$.

Оценка осуществлялась по формуле, подробно обсуждавшейся в [1]:

$$\|\Delta x\| = cond(A)\frac{\|x^0\|\|\Delta b\|}{\|b\|}, \tag{10}$$

где $cond(A) = \|A\|\|A^{-1}\|$ — число обусловленности матрицы A, $\|x^0\|$ — норма вектора точного решения, $\|b\|$ — норма вектора свободных членов.

Для системы (2) при $n = \{5, 10, 15\}$ были получены следующие значения для чисел обусловленности матрицы A: $cond(A) = \{1.2590, 1.5389, 1.9475\}$. Число обусловленности вычислялось с помощью встроенной функции математического пакета Matlab. При заданной точности параметра α $\delta\alpha = 5\%$ возмущение системы $\|\Delta b\|$ для значений $n = \{5, 10, 15\}$ соответственно равнялось $\|\Delta b\| = \{0.0016, 0.0079, 0.0207\}$. Таким образом, значения полученных ошибок по x равны соответственно $\|\Delta x\| = \{0.0020, 0.0133, 0.0454\}$.

3.2 Итерационные методы при неточных входных данных

Кроме того, была проведена оценка числа j_0, которое представляет собой номер итерации процесса (4), на которой нужно его оборвать, чтобы не ухудшить окончательный результат. Действительно, на практике вместо системы (1) мы, обычно имеем дело с системой

$$A_h x = b_h, \tag{11}$$

где A_h и b_h — известные нам приближения матрицы A и вектора b соответственно (т.е. заданные с некой априорной погрешностью). Естественно предположить, что при неточно известных входных данных последовательные приближения имеет смысл вычислять только до тех пор, пока ошибка итерационного процесса не станет сравнимой с априорной неустранимой ошибкой решения системы (11), появляющейся из-за неточности входных данных. В противном случае, результат заведомо не улучшится, но может и существенно ухудшиться. Т.к. в нашем случае матрица коэффициентов A считалась заданной точно, то ограничимся лишь погрешностью по b ($\|\Delta b\| = \|b - b_h\|$), обозначив её согласно [1] $\eta(h) : \|\Delta b\| \leq \eta(h)$. Используя полученную в [1] формулу

$$j_0 = \frac{1}{\ln q} \ln \frac{\eta(h)}{\|b_h\|} \tag{12}$$

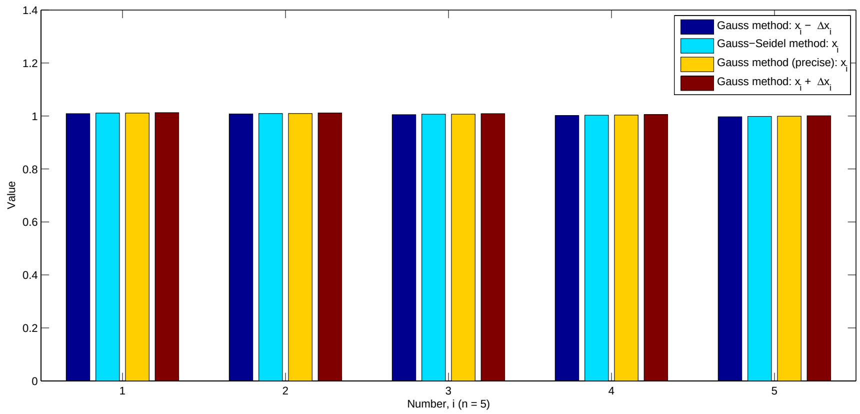
и руководствуясь тем, что принятое для её вывода условие

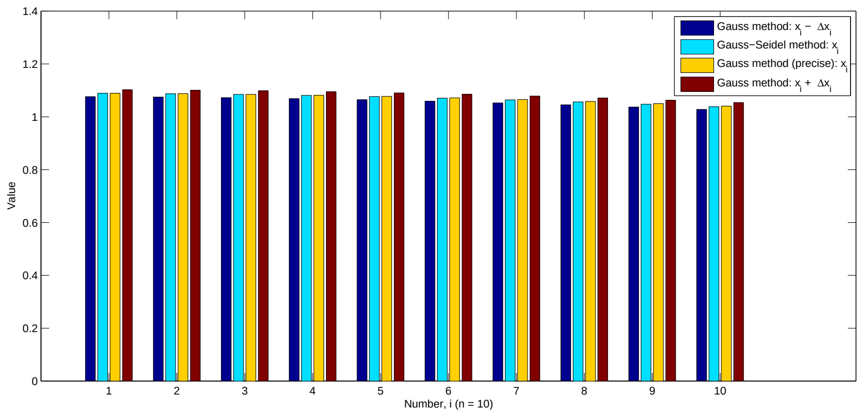
$$q = \|E - A_h\| < 1, \tag{13}$$

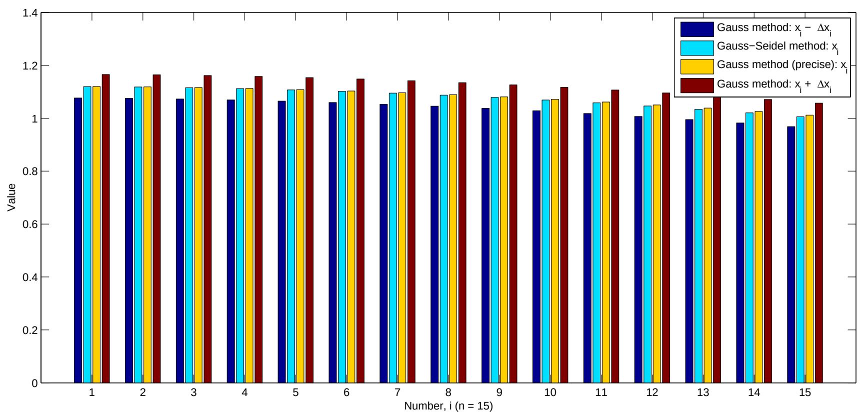
обеспечивающие сходимость, выполнено по крайней мере для наших проверенных значениях параметров (при $n = \{5, 10, 15\}$ соответствующие значения q были $q = \{0.229604, 0.416005, 0.639124\}$) было рассчитано оптимальное число итераций j_0 с учетом того, что $\delta\alpha = 5\%$ (см. соответствующие значения $\|\Delta b\|$ выше): $j_0 = \{5, 7, 12\}$. При этом реальное количество итераций j_{real}, потребовавшееся для достижения заданной точности $\varepsilon = 10^{-5}$ оказалось меньше оценки: $j_{real} = \{4, 5, 6\} < j_0$, что говорит об адекватности выбранной точности решения системы при заданном возмущении.

На рисунках (1), (2) и (3) представлены гистограммы, дающие графическое представление о решении системы (2) для $n = 5$, $n = 10$ и $n = 15$ соответственно. На графиках изображены (в порядке расположения баров) максимальная отрицательная ошибка $(x^0 - \Delta x)$, итерационное решение, полученное методом Гаусса-Зейделя, точное решение, полученное прямым методом Гаусса и, наконец, максимальная положительная ошибка $(x^0 + \Delta x)$. Хорошо видно, что итерационное решение практически совпадает с точным, и, кроме того, оно укладывается в обозначенный коридор ошибок.

4 Результаты расчетов







5 Заключение

Из приведенных выше решений следует, что оба рассмотренных метода хорошо находят решение заданной системы. Преимущество прямого метода Гаусса состоит в простоте его реализации, однако для некоторых больших систем он может оказаться неприемлемым из-за возможных накоплений погрешности и большого расхода ресурсов (неоптимизированности под конкретную систему). Итерационный метод Гаусса-Зейделя реализуется сложнее, однако обеспечивает быструю сходимость к решению системы с заданной точностью, учитывая погрешность входных данных.

Таким образом, оба вышеперечисленных метода могут использоваться для нахождения решения в рамках поставленной нами задачи.

6 Приложения

В приложениях приводятся исходные тексты программы, написанной на языке Matlab. Программа построена из двух модулей – seidel.m и body.m. Главная часть программы — seidel.m — инициализирует параметры, вызывает необходимое число раз модуль body.m, который занимается непосредственно решением системы уравнений выбранным методом, и, наконец, оценивает по полученным из body.m данным ошибки и строит гистограммы.

6.1 seidel.m

```
% Gauss - Seidel solver

clear all

alpha = -pi/100;
dalpha = 0.05;
alpha_l = alpha * (1 - dalpha);
alpha_u = alpha * (1 + dalpha);
n_min = 5;
n_max = 15;
n_step = 5;  % Step

eps = 1e-5; % Precision

n = n_min;

while (n <= n_max)

    % Meth:
```

```
% 1 - solve with G-S (iterative)
% 2 - solve with G (precise)
% 3 - only generate system

fprintf('--- Gauss-Seidel Solver, n = %g ---\n', n);

ru  = body(n, alpha_u, eps, 0,  3);
rl  = body(n, alpha_l, eps, 0, 3);
rc0 = body(n, alpha, eps, 0, 3);

xu = ru(:,1); xl = rl(:,1);
bc = rc0(:,2); bu = ru(:,2); bl = rl(:,2);

dbu = norm(bc - bu);
dbl = norm(bc - bl);
db = max([dbu  dbl]);

rc = body(n, alpha * (1 + dalpha/2), eps, db, 1);
rp = body(n, alpha, eps, db, 2);

xc = rc(:,1);
xp = rp(:,1);

conda = norm(rc(:,3))

dxu = norm(conda * dbu / norm(bc) * norm(xp));
dxl = norm(conda * dbl / norm(bc) * norm(xp));
dx = max([dxu  dxl]);
%db / norm(b)
%norm(xn)

db
dx

xu = xp + dxu;
xl = xp - dxl;

br = [xl, xc, xp, xu];

nf = (n - n_min) / n_step  + 1
f = figure(nf);
close(f);
f = figure(nf);
```

```
    bar(br);
    legend('Gauss method: x_i - \Delta{x_i}','Gauss-Seidel
       method: x_i','Gauss method (precise): x_i','Gauss method
       : x_i + \Delta{x_i}',1);
    xtext = sprintf('Number, i (n = %g)', n);
    xlabel(xtext);
    ylabel('Value');

    n = n + n_step;
end
```

6.2 body.m

```
% Body

function [r] = body(n, alpha, eps, db, meth)

    disp('Generating system...');

    a = zeros(n, n);
    l = a; d = a; r = a;

    for i = 1:1:n
        a(i, i) = 1;
        for j = 1:1:n
            a(i, j) = a(i, j) - sin(j) / 17;
        end
        b(i) = cos(alpha * i);
    end

    b = b';
    a;

    disp('System generated!');



    if (meth ~= 3)
        disp('Solving using standard Gauss algorithm...');
        xn = inv(a) * b;
        xn;

        disp('Solved! Now building matrices...');
```

```
    % Build matrices
    for i = 1:1:n
        for j = 1:1:n
            elm = a(i, j);
            if i == j
                d(i, j) = elm;
            end
            if i > j
                l(i, j) = elm;
            end
            if i < j
                r(i, j) = elm;
            end
        end
    end

    %l
    %d
    %r

    disp('Matrices L, D, R generated!');
    disp('Running Gauss-Seidel algorithm...');

    E = eye(n);
    q = norm(E - a);
    j0 = 1 / log (q) * log ( db / norm(b));

    fprintf('Solving will supposedely take %g iterations.\n
       ', round(j0));
    fprintf('Convergence factor q: %f\n', q);

    xk = b; xkn = b; k = eps * 2; ik = 0;% Intialization

    tk = -inv(d + l) * r;
    ntk = norm(tk)

    while ((k > eps) && (ik < j0))
        % Matrix-based form (debug)
%           hk = inv(d + l);
%           xkn =  tk * xk  + hk * b;

        % Coordinate-based form (production)
        for i = 1:n
```

```
                xkn(i) = b(i);
                if i ~= 1;
                    for j = 1 : i - 1;
                        xkn(i) = xkn(i) - a(i,j) .* xkn(j);
                    end
                end
                if i ~= n;
                    for j = i + 1 : n;
                        xkn(i) = xkn(i) - a(i,j) .* xk(j);
                    end
                end
                xkn(i) = xkn(i) / a(i, i);
            end

            k = norm(xk - xkn) * norm(tk)/ (1 - norm(tk));

            xk = xkn; ik = ik + 1;
        end

        fprintf('OK! Iterations: %g\n', ik);

        xk;
    else
        xz = zeros(n, 1);
    end

    conda = zeros(n, 1); conda(1, 1) = cond(a);

    if (meth == 1)
        r = [xk b conda];
    elseif (meth == 2)
        r = [xn b conda];
    elseif (meth == 3)
        r = [xz b conda];
    end


end
```

Список литературы

[1] Кулинич В.В., Сумин В.И. Решение систем линейных алгебраических уравнений. Введение в численные методы. Лабораторная работа для студентов радиофизического факультета ННГУ. Н. Новгород: изд. ННГУ им. Лобачевского, 1999 г. 28 с.

[2] Калиткин Н.Н., Численные методы. М.: Наука. 1978. 512 с.

[3] Самарский А. А., Введение в численные методы. М.: Наука. 1987. 288 с.

[4] Фаддеев Д. К., Фаддеева В. Н., Вычислительные методы линейной алгебры. М.-Л.: Физматгиз. 1960. 656 с.