Лабораторная работа №3

Решение систем линейных алгебраических уравнений итерационными методами

Цель работы: изучение задачи численного решения систем линейных алгебраических уравнений (СЛАУ); приобретение навыков программирования итерационных методов решения СЛАУ; приобретение навыков использования стандартных средств системы Matlab для решения СЛАУ.

Теоретические сведения

Системой линейных алгебраических уравнений называется система уравнений вида

$$\begin{aligned} &a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \ldots + a_{1n}x_n = b_1, \\ &a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \ldots + a_{2n}x_n = b_2, \\ &\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ &a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \ldots + a_{nn}x_n = b_n. \end{aligned} \tag{1}$$

Ее можно представить в матричном виде $\mathbf{Ax} = \mathbf{b}$, где

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & \ldots & a_{1n} \\ a_{21} & a_{22} & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{32} & a_{33} & \ldots & a_{3n} \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n2} & a_{n3} & \ldots & a_{nn} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} b_1 \\ b_2 \\ b_3 \\ \vdots \\ b_n \end{bmatrix}, \qquad \mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_n \end{bmatrix},$$

при этом предполагается, что $\det \mathbf{A} \neq 0$. Решением СЛАУ называется такой вектор **x**, который при подстановке в (1) превращает каждое уравнение в верное числовое равенство.

Пусть $\mathbf{x}^* = (x_1, \ldots x_n)^T$ - приближенное решение системы (1), тогда вектор $\mathbf{e} = \mathbf{x} - \mathbf{x}^*$ называется погрешностью решения системы уравнений. Часто погрешности решения системы уравнений оценивают по вектору $\mathbf{r} = \mathbf{b} - \mathbf{Ax}^*$, называемому невязкой. Вектор $\mathbf{r}$ показывает, насколько отличается правая часть системы от левой, если подставить в нее приближенное решение. Очевидно, что погрешность и невязка решения должны быть как можно меньше. Величины погрешностей и невязок оцениваются при помощи нормы векторов и матриц.

Нормой вектора $\mathbf{x}$ называется вещественное число $\|\mathbf{x}\|$, обладающее следующими свойствами:

1. $\|\mathbf{x}\| \geq 0$, причем $\|\mathbf{x}\| = 0$ тогда и только тогда, когда $\mathbf{x} = 0$;
2. $\|a\mathbf{x}\| = |a|\,\|\mathbf{x}\|$ для любого вектора $\mathbf{x}$ и любого числа a;
3. $\|\mathbf{x} + \mathbf{y}\| \leq \|\mathbf{x}\| + \|\mathbf{y}\|$ для любых векторов $\mathbf{x}$ и $\mathbf{y}$ (неравенство треугольника).

Норма матрицы $\mathbf{A}$, подчиненная норме векторов $\mathbf{x}$, определяется величиной

$$\|\mathbf{A}\| = \max_{\mathbf{x} \neq \mathbf{0}} \frac{\|\mathbf{Ax}\|}{\|\mathbf{x}\|}.$$

Норма матрицы обладает теми же свойствами, что и норма вектора:

1. $\|\mathbf{A}\| \geq 0$, причем $\|\mathbf{A}\| = 0$ тогда и только тогда, когда $\mathbf{A} = 0$;
2. $\|a\mathbf{A}\| = |a| \|\mathbf{A}\|$ для любой матрицы $\mathbf{A}$ и любого числа a;
3. $\|\mathbf{A} + \mathbf{B}\| \leq \|\mathbf{A}\| + \|\mathbf{B}\|$ для любых матриц $\mathbf{A}$ и $\mathbf{B}$;
4. $\|\mathbf{A} \cdot \mathbf{B}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{B}\|$ для любых матриц $\mathbf{A}$ и $\mathbf{B}$, которые можно умножать;
5. $\|\mathbf{A} \cdot \mathbf{x}\| \leq \|\mathbf{A}\| \cdot \|\mathbf{x}\|$ для любой матрицы $\mathbf{A}$ и любого вектора $\mathbf{x}$.

Существует множество способов введения норм векторов и матриц, однако в вычислительных методах наиболее употребительными являются следующие три:

$$\|\mathbf{x}\|_1 = \sum_{i=1}^{n} |x_i|, \qquad \|\mathbf{x}\|_2 = \left(\sum_{i=1}^{n} |x_i|^2 \right)^{1/2}, \qquad \|\mathbf{x}\|_\infty = \max_{1 \leq i \leq n} |x_i|,$$

$$\|\mathbf{A}\|_1 = \max_{1 \leq j \leq n} \sum_{i=1}^{n} |a_{ij}|, \qquad \|\mathbf{A}\|_2 = \max_{1 \leq j \leq n} \sqrt{\lambda_j(\mathbf{A}^T \mathbf{A})}, \qquad \|\mathbf{A}\|_\infty = \max_{1 \leq i \leq n} \sum_{j=1}^{n} |a_{ij}|,$$

где $\lambda_j(\mathbf{A}^T \mathbf{A})$ - собственные числа матрицы $\mathbf{A}^T \mathbf{A}$.

Одним из важнейших вопросов решения СЛАУ является определение критерия сходимости решения. Эта проблема решается при помощи выработки критерия сходимости по норме. Пусть $\left\{\mathbf{x}^{(k)}\right\}_{k=1}^{\infty}$ - последовательность векторов $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)})$. Последовательность векторов $\mathbf{x}^{(k)}$ сходится к вектору $\mathbf{x}$ при $k \to \infty$ ($\lim\limits_{k \to \infty} \mathbf{x}^{(k)} = \mathbf{x}$), если $\left\|\Delta \mathbf{x}^{(k)}\right\| = \left\|\mathbf{x}^{(k)} - \mathbf{x}\right\| \to 0$ при $k \to \infty$ ($\lim\limits_{k \to \infty} \Delta \mathbf{x}^{(k)} = 0$).

Если система линейных уравнений определена приближенно, т.е. имеются некоторые малые возмущения как коэффициентов a_{ij}, так и коэффициентов b_i, то относительная погрешность решения определяется из следующего неравенства:

$$\frac{\|\Delta \mathbf{x}\|}{\|\mathbf{x}\|} \leq \frac{\nu(\mathbf{A})}{1 - \nu(\mathbf{A}) \dfrac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|}} \left(\frac{\|\Delta \mathbf{A}\|}{\|\mathbf{A}\|} + \frac{\|\Delta \mathbf{b}\|}{\|\mathbf{b}\|} \right),$$

где $\nu(\mathbf{A}) = \left\|\mathbf{A}^{-1}\right\| \|\mathbf{A}\|$ - *число обусловленности* матрицы $\mathbf{A}$, $\Delta \mathbf{A} = \mathbf{A}^* - \mathbf{A}$; $\Delta \mathbf{x} = \mathbf{x}^* - \mathbf{x}$; $\Delta \mathbf{b} = \mathbf{b}^* - \mathbf{b}$. (Звездочкой обозначены приближенные значения.) При $\nu(\mathbf{A}) \gg 1$ система (1) *плохо обусловлена* и следует ожидать больших отклонений предложенного решения от точного.

Метод простой итерации решения СЛАУ. Для использования этого метода исходное уравнение $\mathbf{A}\mathbf{x} = \mathbf{b}$ преобразуется к виду

$$\mathbf{x} = \mathbf{B}\mathbf{x} + \mathbf{c}. \tag{2}$$

Процесс вычисления решения начинается с выбора начального приближения $\mathbf{x}^{(0)} = (x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)})^T$. Подставляя его в правую часть системы (2) и вычисляя полученное выражение, находим первое приближение $\mathbf{x}^{(1)}$, подставляя его аналогичным образом в уравнение (2), получаем второе приближение $\mathbf{x}^{(2)}$. Продолжая этот процесс, получаем последовательность приближений $\left\{\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(k)}, \ldots\right\}$, вычисляемых по формуле

$$\mathbf{x}^{(k+1)} = \mathbf{B}\mathbf{x}^{(k)} + \mathbf{c}.$$

Сходимость метода простой итерации определяется следующими утверждениями. При выполнении условия $\|\mathbf{B}\|<1$ справедливо: 1) решение системы (2) существует и единственно; 2) при произвольном начальном приближении $\mathbf{x}^{(0)}$ метод простой итерации сходится и справедлива оценка погрешности $\left\|\mathbf{x}^{(n)}-\mathbf{x}^*\right\| \le \|\mathbf{B}\|^n \cdot \left\|\mathbf{x}^{(0)}-\mathbf{x}^*\right\|$, где $\mathbf{x}^*$ **-** точное решение.

Для сходимости метода простой итерации достаточно, чтобы матрица $\mathbf{A}$ была близка к матрице с преобладанием диагональных элементов $\left(\sum_{i=1(i\ne j)}^{m} |a_{ij}| < |a_{jj}| \text{ или } \sum_{j=1(j\ne i)}^{m} |a_{ij}| < |a_{ii}| \right)$. Из оценки погрешности следует, что при выполнении условия $\|\mathbf{B}\|<1$ метод простой итерации сходится со скоростью геометрической прогрессии, знаменатель которой $q=\|\mathbf{B}\|$. Скорость сходимости тем выше, чем меньше величина $\|\mathbf{B}\|$. Критерием окончания итерационного процесса (апостериорной оценкой погрешности) выбирают условие $\frac{\|\mathbf{B}\|}{1-\|\mathbf{B}\|}\left\|(\mathbf{x}^{(n)}-\mathbf{x}^{(n-1)})\right\| \le \varepsilon$.

Метод Зейделя. Основная идея метода состоит в том, что при вычислении очередного $(k+1)$-го приближения к неизвестному x_i при $i>1$ используются уже найденные $(k+1)$-е приближения к неизвестным $x_1, x_2, ..., x_{i-1}$, а не k-е приближение, как в методе простой итерации. На $(k+1)$-й итерации компоненты вычисляются по формулам

$$\begin{aligned}
x_1^{(k+1)} &= \qquad\quad + b_{12}x_2^{(k)} + ... + b_{1n}x_n^{(k)} + c_1, \\
x_2^{(k+1)} &= b_{21}x_1^{(k+1)} \qquad\quad + ... + b_{2n}x_n^{(k)} + c_2, \\
&. \; . \\
x_n^{(k+1)} &= b_{n1}x_1^{(k+1)} + b_{n2}x_2^{(k+1)} + ... \qquad + c_n.
\end{aligned} \tag{3}$$

Введем верхнюю и нижнюю строго треугольные матрицы

$$B_L = \begin{bmatrix} 0 & 0 & 0 & ... & 0 \\ b_{21} & 0 & 0 & ... & 0 \\ b_{31} & b_{32} & 0 & ... & 0 \\ . & . & . & ... & . \\ b_{n1} & b_{n2} & b_{n3} & ... & 0 \end{bmatrix}, \quad B_U = \begin{bmatrix} 0 & b_{12} & b_{13} & ... & b_{1n} \\ 0 & 0 & b_{23} & ... & b_{2n} \\ 0 & 0 & 0 & ... & b_{3n} \\ . & . & . & ... & . \\ 0 & 0 & 0 & ... & 0 \end{bmatrix}.$$

Тогда система (3) примет компактный вид:

$$\mathbf{x}^{(k+1)} = \mathbf{B}_L\mathbf{x}^{(k+1)} + \mathbf{B}_U\mathbf{x}^{(k)} + \mathbf{c}\,. \tag{4}$$

Как и для любого итерационного процесса при использовании метода Зейделя возникают два вопроса: каковы достаточные условия сходимости и каков критерий окончания итерационного процесса. При выполнении условия $\|\mathbf{B}_L\|+\|\mathbf{B}_U\|<1$ метод Зейделя сходится при любом выборе начального приближения и верна оценка погрешности $\left\|\mathbf{x}^{(n)}-\mathbf{x}^*\right\| \le q^n \cdot \left\|\mathbf{x}^{(0)}-\mathbf{x}^*\right\|$, где $q=\|\mathbf{B}_U\| / (1-\|\mathbf{B}_L\|)<1$. Если требуется найти решение системы (4) с точностью ε, то итерации по методу Зейделя следует вести до выполнения неравенства $\left\|\mathbf{x}^{(n)}-\mathbf{x}^{(n-1)}\right\|\frac{\|\mathbf{B}_U\|}{1-\|\mathbf{B}\|}\le\varepsilon$, что является критерием окончания итерационного процесса.

Средства Matlab для решения СЛАУ. Для организации вычислительного процесса при выполнении работы потребуются следующие операторы Matlab.

1. Оператор L=**tril**(A) сохраняет нижнюю треугольную часть матрицы **A**. Команда L=**tril**(A,*k*) сохраняет нижнюю часть матрицы **A**, начиная с диагонали с номером k (при $k = 0$ - это главная диагональ, при $k > 0$ - выше главной диагонали и при $k < 0$ - ниже).
2. Оператор L=**triu**(A) сохраняет верхнюю треугольную часть матрицы **A**. Команда L=**triu**(A,*k*) сохраняет верхнюю часть матрицы **A**, начиная с диагонали с номером k (при $k = 0$ - это главная диагональ, при $k > 0$ - выше главной диагонали и при $k < 0$ - ниже).
3. Если **b** - вектор-столбец или вектор-строка размера n, то оператор R=**diag**(b,k) создает квадратную матрицу размерности n+abs(k) с элементами вектора **b** на k-й диагонали. При $k = 0$ - это главная диагональ, при $k > 0$ - выше главной диагонали и при $k < 0$ - ниже. Оператор R=**diag**(b) помещает компоненты вектора **b** на главную диагональ. Если **A** - матрица, то R=**diag**(A, k) - вектор-столбец, составленный из элементов k-й диагонали матрицы **A**. R=**diag**(A) - вектор-столбец с элементами главной диагонали матрицы **A**. Поэтому R=**diag**(**diag**(A)) - диагональная матрица.
4. Обратная матрица для матрицы **A** вычисляется оператором **inv(A)** или **A^-1**.
5. Евклидову норму можно найти с помощью оператора **norm**(b) для вектора b и **norm**(A,'fro') - норма Фробениуса (эвклидова норма матрицы) для матрицы A. **norm**(A,1) определяет столбцовую норму матрицы, **norm**(A,2), **norm**(A,inf) определяют строчную норму; **norm**(A) - так называемая спектральная норма.
6. Оператор **d=eig**(A) вычисляет собственные значения матрицы **A**.

Для получения точного решения используются выражения x=**inv**(A)*b либо x=A^–1*b, либо x=A\b. Запись A\B означает левое деление матрицы **A** на матрицу **B**. По смыслу это то же, что и **inv**(A)*B, однако расчеты выполняются по-другому: запись x=A\b означает решение СЛАУ методом исключения Гаусса.

При работе с матрицами могут быть полезны следующие операторы.

7. Оператор R=**zeros**(n) создает матрицу **R** размерности n ∗ n состоящую из нулей. Команда R=**zeros**(m,n) создает матрицу нулей размера m ∗ n. Оператор R=**zeros**(size(A)) образует матрицу нулей такой же размерности, как и матрица **A**.
8. Оператор R=**ones**(n) создает матрицу единиц размерности n ∗ n. Оператор R=**ones**(m,n) создает матрицу единиц размерности m ∗ n. Оператор R=**zeros**(**size**(A)) образует матрицу единиц такой же размерности, как и матрица **A**.
9. Оператор R=**eye**(n) создает матрицу размерности n ∗ n с единицами на главной диагонали. Оператор R=**eye**(**size**(A)) образует матрицу такой же размерности, как и матрица **A**.
10. Ранг матрицы **A** и ее определитель вычисляются операторами **rank**(A) и **det**(A) соответственно.
11. Оператор **cond**(A) вычисляет число обусловленности матрицы **A** в спектральной норме. Оператор **cond**(A,p) вычисляет число обусловленности по **p**-нормам.

Порядок выполнения работы

1. С помощью программы сгенерировать матрицу **A** и вектор правых частей b для СЛАУ вида **Ax=b**.
2. Проверить условия сходимости итерационных методов для матрицы **A**. При необходимости преобразовать матрицу **A** к виду, позволяющему вычислить решения.
3. Вычислить решения с помощью метода простой итерации и Зейделя, количество итерации $n = 20$. Построить графики зависимости x_i от n. Вычислить вектор невязки решения и ее норму.

4. Получить точное решение СЛАУ, вычислить вектор погрешности решений по методам простой итерации Зейделя и их нормы.

5. Оценить относительную погрешность решения СЛАУ, если $\Delta\mathbf{A}$ определяется десятипроцентным увеличением диагональных элементов, а $\Delta\mathbf{b}$ таким же уменьшением всех элементов.

Содержание отчета

Отчет должен содержать:

1) цель работы, задание;
2) описание метода решения, краткие сведения из теории (формулы, алгоритм и т.п.);
3) программу (распечатку), ее описание;
4) сравнение результатов расчета;
5) краткие выводы.

Задание

Программа генерации матрицы A и вектора b:

```
n1=input('Введите номер группы: ');
n2=input('Введите ваш номер в списке группы: ');
n=n1+n2;
rand('seed',n);
A=rand(7);
b=10*rand(7,1);
Au=triu(A);
Al=tril(A);
Ad=diag(diag(A));
su=sum(sum(abs(Au)));
sl=sum(sum(abs(Al)));
Su=sum(abs(Au));
Sl=sum(abs(Al));
sigma=5;
O=ones(7);
Ou=triu(O);
S6=diag(sigma*(Su./Sl));
At=sigma*((Al+Ou)./(Au'+Ou));
neo=rem(n2,2);
mode=rem(n2,3);
if (mode==0) Al=(sigma*su/sl)*Al; end
if (mode==1) Al=S6*Al; end
if (mode==2) Al=At.*Al; end
A=Al+Ad+Au;
ma=max(max(abs(A)));
if (ma>1000) A=0.08*A; end
A=0.5*A+2.75*diag(diag(A)), b
```

Контрольные вопросы

1. Дайте определение погрешности и невязки решения СЛАУ. Как они связаны друг с другом?

2. Дайте определение нормы вектора и способов ее вычисления. Приведите примеры.
3. Дайте определение нормы матрицы и способов ее вычисления. Приведите примеры матричных норм, реализованных в пакете Matlab.
4. Опишите алгоритм метода Гаусса решения СЛАУ.
5. Опишите алгоритм метода Гаусса для решения СЛАУ с трехдиагональной матрицей (метод прогонки).
6. Сформулируйте условия корректности и устойчивости метода прогонки.
7. Сформулируйте неравенство, определяющее относительную погрешность решения СЛАУ.
8. Что такое число обусловленности матрицы?
9. Опишите алгоритм решения системы методом Якоби.
10. Опишите алгоритм Зейделя.
11. Сформулируйте априорную оценку погрешности метода простой итерации.
12. Сформулируйте априорную оценку погрешности метода Зейделя.
13. Сформулируйте апостериорную оценку погрешности метода простой итерации.
14. Сформулируйте критерий окончания итерационного процесса метода Зейделя.
15. Сформулируйте условия сходимости метода простой итерации.
16. Сформулируйте условия сходимости метода Зейделя.

Литература

Гончаров В.А., Земсков В.Н., Яковлев В.Б. Лабораторный практикум по курсу «Вычислительная математика». – М.: МИЭТ, 2008.