

Собственные значения и собственные векторы

Одна из важных задач вычислительной линейной алгебры – нахождение собственных чисел и собственных векторов матрицы. Прежде чем перейти к изучению методов её решения, сделаем некоторые вводные замечания.

Пусть \mathbf{A} – матрица с размерностью $n \times n$, λ – некоторое число. Любой ненулевой вектор \mathbf{x} , принадлежащий некоторому векторному пространству, для которого $\mathbf{A} \cdot \mathbf{x} = \lambda \cdot \mathbf{x}$, называется *собственным вектором* матрицы \mathbf{A} , а λ – принадлежащим ему или соответствующим ему *собственным значением* матрицы \mathbf{A} .

Уравнение $\mathbf{A} \cdot \mathbf{x} = \lambda \cdot \mathbf{x}$ эквивалентно уравнению $(\mathbf{A} - \lambda \cdot \mathbf{E})\mathbf{x} = \mathbf{0}$. Это однородная система линейных уравнений, нетривиальные решения которой являются искомыми собственными векторами \mathbf{x} . Она имеет нетривиальные решения только тогда, когда $\text{rang}(\mathbf{A} - \lambda \cdot \mathbf{E}) < n$, то есть если $\det(\mathbf{A} - \lambda \cdot \mathbf{E}) = 0$

Многочлен **$\det (\mathbf{A} - \lambda \cdot \mathbf{E})$** называется *характеристическим многочленом* матрицы **\mathbf{A}** , а уравнение **$\det (\mathbf{A} - \lambda \cdot \mathbf{E}) = 0$** – *характеристическим уравнением* матрицы **\mathbf{A}** или *вековым уравнением*. Если λ_i - собственные значения **\mathbf{A}** , то нетривиальные решения однородной системы линейных уравнений **$(\mathbf{A} - \lambda \cdot \mathbf{E}) \mathbf{x} = 0$** есть собственные векторы матрицы **\mathbf{A}** , принадлежащие собственному значению λ_i .

Множество решений этой системы уравнений называют *собственным подпространством* матрицы **\mathbf{A}** , принадлежащим собственному значению λ_i , каждый ненулевой вектор собственного подпространства является собственным вектором матрицы **\mathbf{A}** .

Теоретически задача нахождения собственных чисел и собственных векторов легко решается: нужно найти корни векового уравнения λ и, подставляя их поочерёдно в систему уравнений:

$$\begin{bmatrix} a_{11} - \lambda & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} - \lambda & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} - \lambda \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}. \quad (3.2.18)$$

находить собственные векторы матрицы **A**.

Практически реализация этого подхода сопряжена с рядом трудностей, возрастающих с ростом размерности решаемой задачи. Эти трудности обусловлены решением векового уравнения и также вычислением корней получающегося многочлена (полинома) n -ой степени λ , а также поиском линейно независимых решений вырожденных СЛАУ - собственных векторов матрицы **A**.

В действительности, из сказанного следует, что характеристическое уравнение (3.2.19) можно переписать в виде равенства нулю характеристического многочлена $P(\lambda)$:

$$P(\lambda) \equiv \det(\overline{A} - \lambda \overline{E}) \equiv \left| \overline{A} - \lambda \overline{E} \right| = (-\lambda)^n + b_1(-\lambda)^{n-1} + \dots + b_{n-1}(-\lambda) + b_n = 0, \quad (3.2.20)$$

где b_1, \dots, b_n — известные коэффициенты получающегося характеристического многочлена.

Отсюда можно сделать вывод, что матрица \overline{A} порядка n имеет n собственных чисел (собственных значений) $\lambda_1, \lambda_2, \dots, \lambda_n$, если каждый корень считать столько раз, какова его кратность.

Рассмотрим пример определения собственных чисел (собственных значений) и собственных векторов следующей матрицы \bar{A} :

$$\bar{A} = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}. \quad (3.2.21)$$

Для этого записывается характеристическое уравнение в виде (3.2.19) или (3.2.20):

$$P(\lambda) = \begin{vmatrix} 1-\lambda & 2 \\ 2 & 1-\lambda \end{vmatrix} = \lambda^2 - 2\lambda - 3 = 0. \quad (3.2.22)$$

Корни этого характеристического уравнения – собственные числа (значения) – соответственно равны: $\lambda_1 = -1$ и $\lambda_2 = 3$.

При нахождении собственного вектора для первого корня ($\lambda_1 = -1$) следует записать однородное СЛАУ (3.2.14):

$$\begin{bmatrix} 1-(-1) & 2 \\ 2 & 1-(-1) \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (3.2.23)$$

в результате чего получается система:

$$\begin{cases} 2x_1 + 2x_2 = 0; \\ 2x_1 + 2x_2 = 0 \end{cases} \quad (3.2.24)$$

или

$$x_1 + x_2 = 0. \quad (3.2.25)$$

Таким образом, для этого собственного числа (собственного значения) $\lambda_1 = -1$ один из собственных векторов (бесчисленного множества решений однородной СЛАУ) может быть представлен как

$$\bar{\xi}_1 = \begin{bmatrix} 1 \\ -1 \end{bmatrix}. \quad (3.2.26)$$

Для второго собственного числа (собственного значения) $\lambda_2 = 3$ по аналогии имеем однородное СЛАУ:

$$\begin{bmatrix} 1-3 & 2 \\ 2 & 1-3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad (3.2.27)$$

что дает

$$\begin{cases} -2x_1 + 2x_2 = 0, \\ 2x_1 - 2x_2 = 0 \end{cases} \quad (3.2.28)$$

или

$$x_1 - x_2 = 0, \quad (3.2.29)$$

откуда второй собственный вектор, соответствующий второму собственному числу $-\lambda_2 = 3$, и являющейся одним из множества решений однородного СЛАУ имеет вид:

$$\bar{\xi}_2 = \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \quad (3.2.30)$$

Оба эти решения $\bar{\xi}_1$ (3.2.26) и $\bar{\xi}_2$ (3.2.30) могут быть умножены на произвольные скалярные множители, в результате и получается множество решений однородных СЛАУ (3.2.24) и (3.2.28).

Рассмотренный непосредственный подход к решению алгебраической проблемы собственных чисел (собственных значений) обычно применяют лишь при очень малых размерностях матриц коэффициентов \bar{A} ($n = 2, 3$); уже при $n \geq 4$ на первый план выходят специальные численные методы решения задач о собственных числах и собственных векторах, в соответствии с которыми разработаны многочисленные алгоритмы, в том числе и для решения однородных СЛАУ.

Иногда требуется найти собственные векторы \mathbf{y} и собственные значения μ , определяемые соотношением $\mathbf{A} \cdot \mathbf{y} = \mu \mathbf{B} \mathbf{y}$ ($\mathbf{y} \neq 0$), где \mathbf{B} – невырожденная матрица. Векторы \mathbf{y} и числа μ обязательно являются собственными векторами и собственными значениями матрицы $\mathbf{B}^{-1}\mathbf{A}$.

Пусть $\mathbf{A} = \{a_{ij}\}$ и $\mathbf{B} = \{b_{ij}\}$, причём матрица \mathbf{B} является положительно определённой, тогда собственные значения μ совпадают с корнями уравнения n -й степени $\det(\mathbf{A} - \mu \mathbf{B}) = \det(a_{ij} - \mu b_{ij}) = 0$. Это уравнение называют *характеристическим уравнением для обобщённой задачи о собственных значениях*. Для каждого корня μ кратности m существует ровно m линейно независимых собственных векторов \mathbf{y} .

Задача на собственные значения может быть легко решена для матриц простой формы: диагональных, трёхдиагональных, треугольных. Определитель треугольной или диагональной матрицы равен произведению диагональных элементов, а матрица $A - \lambda \cdot E$ также является треугольной или диагональной, поэтому собственные значения равны диагональным элементам треугольной или диагональной матрицы. Диагональная матрица имеет n собственных векторов $e_i = \{0, \dots, 0, 1, 0, \dots, 0\}^T$, $i = 1, \dots, n$, которым соответствуют n собственных значений $\lambda_i = a_{ii}$, $i = 1, \dots, n$.

Многие численные методы решения задач на собственные значения основаны на приведение матрицы к одной из перечисленных выше форм с помощью *преобразования подобия*.

Матрица **B** называется *подобной* матрице **A**, если существует обратимая матрица **F** такая, что **B = F⁻¹AF**. Преобразовав исходное равенство **A·x = λ·x**, приведём к виду:

$$A \mathbf{F} \mathbf{F}^{-1} \mathbf{x} = \lambda \mathbf{F} \mathbf{F}^{-1} \mathbf{x} \qquad \mathbf{F}^{-1} \mathbf{A} \mathbf{F} \mathbf{F}^{-1} \mathbf{x} = \lambda \mathbf{F}^{-1} \mathbf{x}$$

$$\mathbf{B} \mathbf{y} = \lambda \mathbf{y} \qquad \mathbf{y} = \mathbf{F}^{-1} \mathbf{x},$$

где **y** – собственный вектор матрицы **B**. Таким образом, при преобразовании подобия собственные значения не меняются, а собственные векторы преобразуются по определённому закону.

Из равенства **A·x = λ·x** и свойств норм векторов и матриц следует:

$$\| \lambda x \| = | \lambda | \cdot \| x \| = \| Ax \| \leq \| A \| \cdot \| x \|, \quad | \lambda | \leq \| A \|, \quad \text{или}$$

$\rho(A) \leq \| A \|$, где $\rho(A) = \max_i | \lambda_i |$ - максимальное по модулю собственное значение или **спектральный радиус** матрицы **A**. Таким образом, за норму матрицы можно принять её спектральный радиус.

Тогда число обусловленности будет равно:

$$\mu(A) = \|A\| \cdot \|A^{-1}\| = \max_i |\lambda_i| \frac{1}{\min |\lambda_i|} \geq 1,$$

поскольку спектральный радиус обратной матрицы A^{-1} равен обратной величине минимального собственного значения исходной матрицы.

Собственные значения и собственные векторы в MATLAB

- **eig(M)** – возвращает вектор собственных значений матрицы M, вызов функции в формате **[V,D]= eig(M)** даст матрицу V, столбцы которой – собственные векторы матрицы M, и диагональную матрицу D, содержащую собственные значения матрицы M.

```
>> M=[3 -2;-4 1]
```

```
>> eig(M) %Собственные значения матрицы M
```

```
ans =
```

```
5
```

```
-1
```

```
>> [V,D]=eig(M)
```

```
V = %Первый столбец – собственный вектор для 5, второй для -1
```

```
0.7071    0.4472
```

```
-0.7071    0.8944
```

D = %Собственные значения матрицы M

5 0

0 -1

>> %Проверка $M*V=V*D$

>>V*D

ans =

3.5355 -0.4472

-3.5355 -0.8944

>>M*V

ans =

3.5355 -0.4472

-3.5355 -0.8944

С иными формами обращения можно познакомиться,
воспользовавшись помощью **help eig**.

Итерационные методы решения систем линейных алгебраических уравнений

или методы последовательных приближений

(позволяют вычислить последовательность $\{u_k\}$,
сходящуюся к решению задачи при $k \rightarrow \infty$).

На практике ограничиваются конечным k в зависимости от требуемой точности. Однако неточность задания правых частей и элементов матрицы A может приводить к значительным погрешностям при вычислении решения системы.

Их часто используют для решения систем со специальными матрицами или слабо заполненными, разреженными матрицами, а также для систем уравнений высокого порядка $n=10^2 - 10^7$.

При решении систем линейных алгебраических уравнений любым из итерационных методов можно выделить три последовательных этапа:

- приведение исходной системы вида $Au = f$ к преобразованной (итерационной) форме
- анализ системы на сходимость метода
- решение преобразованной системы с точностью ε одним из методов последовательных приближений, в основе которых лежит та или иная стратегия получения вектора решений по его предыдущему приближению.

Метод простых итераций

Рассмотрим систему линейных алгебраических уравнений

$$\mathbf{A}\mathbf{u} = \mathbf{f}.$$

Проведем несколько равносильных преобразований. Умножим обе части системы на один и тот же скалярный множитель τ , затем прибавим к правой и левой частям системы вектор \mathbf{u} . Систему уравнений можно теперь записать в виде, удобном для итераций:

$$\mathbf{u} = \mathbf{B}\mathbf{u} + \mathbf{F}, \tag{2.15}$$

где $\mathbf{B} = \mathbf{E} - \tau\mathbf{A}$, $\mathbf{F} = \tau\mathbf{f}$.

Теперь построим последовательность приближений к решению системы. Выберем произвольный вектор \mathbf{u}_0 — начальное приближение к решению. Чаще всего его просто полагают нулевым вектором. Скорее всего, начальное приближение не удовлетворяет (2.15) и, следовательно, исходной системе. При подстановке его в исходное уравнение возникает невязка $\mathbf{r}_0 = \mathbf{f} - \mathbf{A}\mathbf{u}_0$. Вычислив невязку, с помощью (2.15) можно уточнить приближение к решению, считая, что

$$\mathbf{u}_1 = \mathbf{u}_0 + \tau \mathbf{r}_0.$$

По первому приближению снова вычисляется невязка, процесс продолжается. В ходе итерации получаем $\mathbf{u}_{k+1} = \mathbf{u}_k + \tau \mathbf{r}_k$, $\mathbf{r}_k = \mathbf{f} - \mathbf{A}\mathbf{u}_k$. Эквивалентная формулировка метода, называемого методом простых итераций, заключается в следующем. Решение (2.15) находится как предел последовательности $\{\mathbf{u}_0, \mathbf{u}_1, \mathbf{u}_2, \dots\}$ приближений, члены которой связаны рекуррентным соотношением (оно эквивалентно приведенному выше, из записи исключен вектор невязки):

$$\mathbf{u}_{k+1} = \mathbf{B}\mathbf{u}_k + \mathbf{F}, \quad (2.16)$$

$\mathbf{u}_0 = \mathbf{0}$ (или любому произвольному вектору). Если предел такой последовательности существует, то говорят о *сходимости* итерационного процесса к решению СЛАУ.

Существуют другие формы записи метода итераций, например

$$\mathbf{u}_{k+1} = (\mathbf{E} - \tau \mathbf{A})\mathbf{u}_k + \tau \mathbf{f}. \quad (2.17)$$

Канонической формой записи двухслойного итерационного процесса называется следующая:

$$\mathbf{D}_{k+1} \frac{\mathbf{u}_{k+1} - \mathbf{u}_k}{\tau_{k+1}} + \mathbf{A}\mathbf{u}_k = \mathbf{f}. \quad (2.18)$$

При $\mathbf{D}_k = \mathbf{E}$, $\tau_k = \tau$ последняя формула соответствует однопараметрическому итерационному процессу — рассмотренному выше *методу простых итераций*. При $\mathbf{D}_k = \mathbf{E}$, $\tau_k = \{\tau_k, k = 1, \dots, n\}$ — n -шаговому явному итерационному процессу, при $\mathbf{D}_k = \mathbf{E}$, $\tau_k = 1$ — методу простой итерации без итерационного параметра. В случае, когда $\mathbf{D} \neq \mathbf{E}$, итерационный метод называется *неявным* — для вычисления следующего приближения к решению придется решать (как правило, более простую, чем исходную) систему линейных уравнений.

Теорема (достаточное условие сходимости метода простой итерации).
Итерационный процесс (2.16) сходится к решению U СЛАУ $\mathbf{A}\mathbf{u} = \mathbf{F}$ со скоростью геометрической прогрессии при выполнении условия: $\|\mathbf{B}\| \leq q < 1$.

Теорема (критерий сходимости метода простой итерации (без доказательства)). Пусть СЛАУ (2.15) имеет единственное решение. Тогда для сходимости итерационного процесса (2.16) необходимо и достаточно, чтобы все собственные значения матрицы B по абсолютной величине были меньше единицы.

Сравним по количеству арифметических действий прямые и итерационные методы. Метод Гаусса без выбора главного элемента при $n \gg 1$ требует $\approx (\frac{2}{3}n^3)$ арифметических действий; метод простой итерации (2.16) $\approx (2n^2 \cdot I)$, где I — число приближений, необходимое для достижения заданной точности. Значит, при $I < n/3$ метод итераций становится предпочтительнее. В реальных задачах, в основном, $I \ll n$. Кроме того, итерационные методы можно делать более эффективными, изменяя итерационные параметры. В ряде случаев итерационные методы оказываются более устойчивыми по отношению к накоплению ошибок округления, чем прямые.

Метод простых итераций

Метод простых итераций — один из наиболее распространенных методов решения СЛАУ, который отличается простотой вычислительного процесса и во многих случаях дает хорошую сходимость.

Представить систему из трех уравнений вида (3.2.9) в итерационной форме можно путем записи каждого его уравнения в виде решения относительно одного из неизвестных, например:

$$\begin{cases} x_1 = & + c_{12}x_2 + c_{13}x_3 + d_1; \\ x_2 = c_{21}x_1 & + c_{23}x_3 + d_2; \\ x_3 = c_{31}x_1 + c_{32}x_2 & + d_3 \end{cases} \tag{3.2.49}$$

или, в матричном виде,

$$\bar{x} = \bar{\bar{C}}\bar{x} + \bar{d}, \tag{3.2.50}$$

где

$$\bar{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}; \quad \bar{\bar{C}} = \begin{bmatrix} 0 & c_{12} & c_{13} \\ c_{21} & 0 & c_{23} \\ c_{31} & c_{32} & 0 \end{bmatrix}; \quad \bar{d} = \begin{bmatrix} d_1 \\ d_2 \\ d_3 \end{bmatrix}.$$

Элементы матрицы $\bar{\bar{C}}$ и вектора \bar{d} вычисляются по формулам:

$$c_{ij} = -a_{ij} / a_{ii}, \quad d_i = b_i / a_{ii}, \quad i, j = 1, 2, 3.$$

При использовании итерационного метода решения необходимо оценить сходимость метода для данной системы, которая зависит только от матрицы коэффициентов $\overline{\overline{C}}$. Процесс сходится в том случае, если норма матрицы $\overline{\overline{C}}$ меньше единицы, т.е.

$$\|\overline{\overline{C}}\| = \sqrt{\sum_{i=1}^3 \sum_{j=1}^3 c_{ij}^2} < 1. \quad (3.2.51)$$

Это условие является достаточным для сходимости метода. Для его выполнения необходимо, чтобы на этапе приведения к итерационной форме (3.2.49) каждое уравнение системы (3.2.9) решалось относительно той неизвестной переменной, которая имеет наибольший по модулю коэффициент. Поэтому порядок расположения уравнений в системе имеет важное значение.

Стратегия метода простых итераций основана на последовательном приближении к искомому решению системы, при этом каждое следующее $(k+1)$ приближение получается в результате подстановки в правую часть преобразованной системы (3.2.49) приближения, полученного на предыдущей k -й итерации, т.е.

$$\overline{x}^{(k+1)} = \overline{\overline{C}} \overline{x}^{(k)} + \overline{d}. \quad (3.2.52)$$

В качестве начального приближения обычно принимают вектор-столбец свободных членов преобразованной системы: $\overline{x}^{(0)} = \overline{d}$. Условие окончания итерационного процесса для получения решения принимает вид

$$|x_i^{(k+1)} - x_i^{(k)}| \leq \varepsilon \quad (i = 1, 2, 3) \quad (3.2.53)$$

или

$$\max_i |x_i^{(k+1)} - x_i^{(k)}| \leq \varepsilon \quad (i = 1, 2, 3). \quad (3.2.54)$$

Пример. Методом простых итераций с точностью $\varepsilon = 0,1$ решить систему

$$\begin{cases} -7x_1 - 2x_2 + x_3 = 1; \\ 8x_1 + 3x_2 + 3x_3 = 2; \\ x_1 + 6x_2 - x_3 = 3. \end{cases} \quad (3.2.55)$$

▼ Для получения преобразованной системы уравнений необходимо сделать допустимые преобразования над исходной системой (3.2.55) с тем, чтобы диагональные коэффициенты матрицы были максимальными по модулю. Для этого второе уравнение сделаем первым и в качестве второго используем третье. Сложив первое и второе уравнения исходной системы, получим третье: $x_1 + x_2 + 4x_3 = 3$.

Тогда в итерационной форме имеем систему

$$\begin{cases} x_1 = -(3/8)x_2 - (3/8)x_3 + 2/8; \\ x_2 = -(1/6)x_1 + (1/6)x_3 + 3/6; \\ x_3 = -(1/4)x_1 - (1/4)x_2 + 3/4. \end{cases}$$

Проверяем условие сходимости (3.2.51):

$$\|C\| = \sqrt{(-3/8)^2 + (-3/8)^2 + (-1/6)^2 + (1/6)^2 + (1/4)^2 + (-1/4)^2} = \sqrt{266/576} < 1;$$

так как $\|C\| < 1$, то решение может быть получено методом простых итераций. Результаты расчета представлены в табл. 3.1.

Таблица 3.1. Значения приближённых решений, получаемых на последовательных итерациях при решении СЛАУ (3.2.55) методом простых итераций

Итерации	x_1	x_2	x_3	$\max_i x_i^{(k+1)} - x_i^{(k)} $
0	0.25	0.5	0.75	
1	-0.125	0.583	0.75	0.13
2	-0.217	0.583	0.575	0.18
3	-0.308	0.632	0.559	0.09

Таким образом, для получения решения с заданной точностью $\varepsilon = 0,1$ потребовались три итерации: $x_1 = x_1^{(3)} = -0.308$, $x_2 = x_2^{(3)} = 0.632$, $x_3 = x_3^{(3)} = 0.559$. ▲

Влияние ошибок округления на результат численного решения

Будем трактовать суммарный эффект ошибок округления при выполнении одного итерационного шага как возмущение правой части в итерационном процессе

$$\mathbf{u}_k = \mathbf{B}\mathbf{u}_{k-1} + \mathbf{F}. \quad (2.19)$$

Результат вычислений на каждой итерации при наличии ошибок округления представим в виде

$$\mathbf{u}_k^M = \mathbf{B}\mathbf{u}_{k-1}^M + \mathbf{F} + \delta_k, \quad (2.20)$$

где δ_k — суммарная погрешность округления. Норму разности между реальным и идеальным (т. е. в отсутствии ошибки округления) результатами

расчетов получим, вычитая (2.19) из (2.20). Учтем, что $\|\mathbf{B}\| < q < 1$,

$$\begin{aligned} \|\mathbf{u}_k^M - \mathbf{u}_k\| &\leq q \|\mathbf{u}_{k-1}^M - \mathbf{u}_{k-1}\| + \\ &+ \|\delta_k\| \leq q^2 \|\mathbf{u}_{k-2}^M - \mathbf{u}_{k-2}\| + q \|\delta_{k-1}\| + \|\delta_k\| \leq \dots \leq q^k \|\mathbf{u}_0^M - \mathbf{u}_0\| + \\ &+ (\max_i \|\delta_i\|)(1 + q + \dots + q^{k-1}), i = 1, \dots, k. \end{aligned}$$

Так как начальное приближение задано точно $\|\mathbf{u}_0^M - \mathbf{u}_0\| = 0$. Обозначим $\delta = \max_i \|\delta_i\|$ и вычислим сумму членов геометрической прогрессии. Получим $\|\mathbf{u}_k^M - \mathbf{u}_k\| \leq \delta \frac{q^k - 1}{q - 1} \leq \frac{\delta}{1 - q}$, то есть погрешность, вносимая в решение из-за конечной разрядности мантииссы, не зависит от количества итераций. Этот результат является характеристикой устойчивости рассматриваемого вычислительного процесса.

Методы Якоби, Зейделя, верхней релаксации

Представим матрицу A в виде

$$A = L + D + U, \quad (2.21)$$

где L и U — нижняя и верхняя треугольные матрицы с нулевыми элементами на главной диагонали, D — диагональная матрица. Рассматриваемая СЛАУ может быть переписана в следующем эквивалентном виде:

$$Lu + Du + Uu = f.$$

Построим два итерационных метода

$$\mathbf{L}\mathbf{u}_k + \mathbf{D}\mathbf{u}_{k+1} + \mathbf{U}\mathbf{u}_k = \mathbf{f} \quad \text{Якоби}$$

и

$$\mathbf{L}\mathbf{u}_{k+1} + \mathbf{D}\mathbf{u}_{k+1} + \mathbf{U}\mathbf{u}_k = \mathbf{f}, \quad \text{Зейделя}$$

или, соответственно,

$$\mathbf{u}_{k+1} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U})\mathbf{u}_k + \mathbf{D}^{-1}\mathbf{f} \quad \text{Якоби} \quad (2.22)$$

и

$$\mathbf{u}_{k+1} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}\mathbf{u}_k + (\mathbf{L} + \mathbf{D})^{-1}\mathbf{f}. \quad \text{Зейделя} \quad (2.23)$$

Очевидно, что эти формулы описывают итерационные процессы вида (2.16), если положить в (2.22)

$$\mathbf{B} = -\mathbf{D}^{-1}(\mathbf{L} + \mathbf{U}), \mathbf{F} = \mathbf{D}^{-1}\mathbf{f}$$

или

$$\mathbf{B} = -(\mathbf{L} + \mathbf{D})^{-1}\mathbf{U}, \mathbf{F} = (\mathbf{L} + \mathbf{D})^{-1}\mathbf{f}.$$

Эти итерационные процессы называются *методами Якоби и Зейделя*. Представим их в компонентной записи. Метод Якоби будет иметь вид (перенесем итерационный индекс k вверх):

$$\begin{aligned} u_1^{k+1} &= -(a_{12}u_2^k + a_{13}u_3^k + \dots + a_{1n}u_n^k - f_1)/a_{11}, \\ u_2^{k+1} &= -(a_{21}u_1^k + a_{23}u_3^k + \dots + a_{2n}u_n^k - f_2)/a_{22}, \\ &\dots \\ u_n^{k+1} &= -(a_{n1}u_1^k + a_{n2}u_2^k + \dots + a_{n,n-1}u_{n-1}^k - f_n)/a_{nn}. \end{aligned}$$

Метод Зейделя можно представить следующим образом:

$$\begin{aligned} u_1^{k+1} &= -(a_{12}u_2^k + a_{13}u_3^k + \dots + a_{1n}u_n^k - f_1)/a_{11}, \\ u_2^{k+1} &= -(a_{21}u_1^{k+1} + a_{23}u_3^k + \dots + a_{2n}u_n^k - f_2)/a_{22}, \\ &\dots \\ u_n^{k+1} &= -(a_{n1}u_1^{k+1} + a_{n2}u_2^{k+1} + \dots + a_{n,n-1}u_{n-1}^{k+1} - f_n)/a_{nn}. \end{aligned}$$

при вычислении компонента $u_i^{(k+1)}$ вектора неизвестных на $(k+1)$ -й итерации используются $u_1^{(k+1)}$, $u_2^{(k+1)}$, \dots , $u_{i-1}^{(k+1)}$, уже вычисленные на $(k+1)$ -й итерации. Значения остальных компонентов $u_i^{(k)}$, $u_{i+1}^{(k)}$, \dots , $u_n^{(k)}$ берутся из предыдущей итерации.

Теорема (достаточное условие сходимости метода Якоби). *Итерационный метод Якоби сходится к решению соответствующей СЛАУ, если выполнено условие диагонального преобладания*

$$|a_{ii}| > \sum_{\substack{j=1 \\ (j \neq i)}}^n |a_{ij}|, i = 1, \dots, n. \quad (2.24)$$

Теорема (критерий сходимости итерационного метода Якоби). *Для сходимости итерационного метода Якоби необходимо и достаточно, чтобы все корни уравнения*

$$\begin{vmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \lambda a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & \lambda a_{nn} \end{vmatrix} = 0$$

по модулю не превосходили единицы.

Аналогичную теорему можно доказать и для метода Зейделя, однако матрица в этой теореме будет иметь другой вид:

$$\begin{pmatrix} \lambda a_{11} & a_{12} & \dots & a_{1n} \\ \lambda a_{12} & \lambda a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ \lambda a_{n1} & \lambda a_{n2} & \dots & \lambda a_{nn} \end{pmatrix}.$$

Если СЛАУ имеет диагональное преобладание, т.е. $|a_{ii}| > \sum_{\substack{j=1 \\ i \neq j}}^n |a_{ij}|$, $i = \overline{1, n}$, методы простых итераций и Зейделя сходятся. Однако единственное решение СЛАУ **$Au = f$** существует и тогда, когда нет диагонального преобладания, но матрица A является невырожденной (т.е. $\det A \neq 0$).

С помощью элементарных преобразований можно из невырожденной матрицы построить матрицу с диагональным преобладанием, для которой рассмотренные итерационные методы будут сходиться. Однако существует более простой способ, а именно нормализовать СЛАУ **$Au = f$** с невырожденной матрицей A и применить к ней метод Зейделя.

СЛАУ называется *нормальной*, если ее матрица симметрична ($A = A^T$) и положительно определена. *Положительно определенная матрица* — матрица, для которой квадратичная форма $(Ax, x) > 0$, или матрица, у которой все собственные значения положительны. По критерию Сильвестра для положительно определенной матрицы все диагональные миноры матрицы положительны:

$$a_{11} > 0, \quad \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} > 0, \quad \dots, \quad \det A > 0.$$

Из любой невырожденной матрицы можно сделать нормальную, домножив ее на транспонированную слева. Умножим $\mathbf{A}\mathbf{u} = \mathbf{f}$ слева на A^T , получим

$$A^T A \mathbf{u} = A^T \mathbf{f} \quad \text{или} \quad B \mathbf{u} = \mathbf{c},$$

где $B = A^T A$ — нормальная матрица, $\mathbf{c} = A^T \mathbf{f}$

Справедлива следующая теорема.

Теорема *Метод Зейделя всегда сходится для нормальных СЛАУ.*

Метод Зейделя обладает более быстрой сходимостью, чем метод простых итераций.

Развитием метода Зейделя является метод релаксации. В этом методе вводится итерационный параметр τ , называемый параметром релаксации. Представим метод релаксации в матричной форме:

$$(\tau \mathbf{L} \mathbf{u}_{k+1} + \mathbf{D} \mathbf{u}_{k+1}) + (\tau - 1) \mathbf{D} \mathbf{u}_k + \tau \mathbf{U} \mathbf{u}_k = \tau \mathbf{f}.$$

Выбирая τ , можно существенно изменять скорость сходимости итерационного метода. Выразим \mathbf{u}_{k+1}

$$\mathbf{u}_{k+1} = -(\mathbf{D} + \tau \mathbf{L})^{-1} [(\tau - 1) \mathbf{D} + \tau \mathbf{L}] \mathbf{u}_k + \tau (\mathbf{D} + \tau \mathbf{L})^{-1} \mathbf{f}.$$

В общем случае задача вычисления $\tau_{\text{опт}}$ (оптимального итерационного параметра) не решена, однако известно, что $1 < \tau_{\text{опт}} < 2$. В этом случае итерационный метод называется методом последовательной верхней релаксации или SOR — Successive Over Relaxation. Иногда встречается термин «сверхрелаксация» при $1 < \tau_{\text{опт}} < 2$. При $0 < \tau < 1$ имеем метод нижней релаксации.

Графическая интерпретация метода Якоби для решения СЛАУ вида:

$$a_{11}u + a_{12}v = f_1,$$

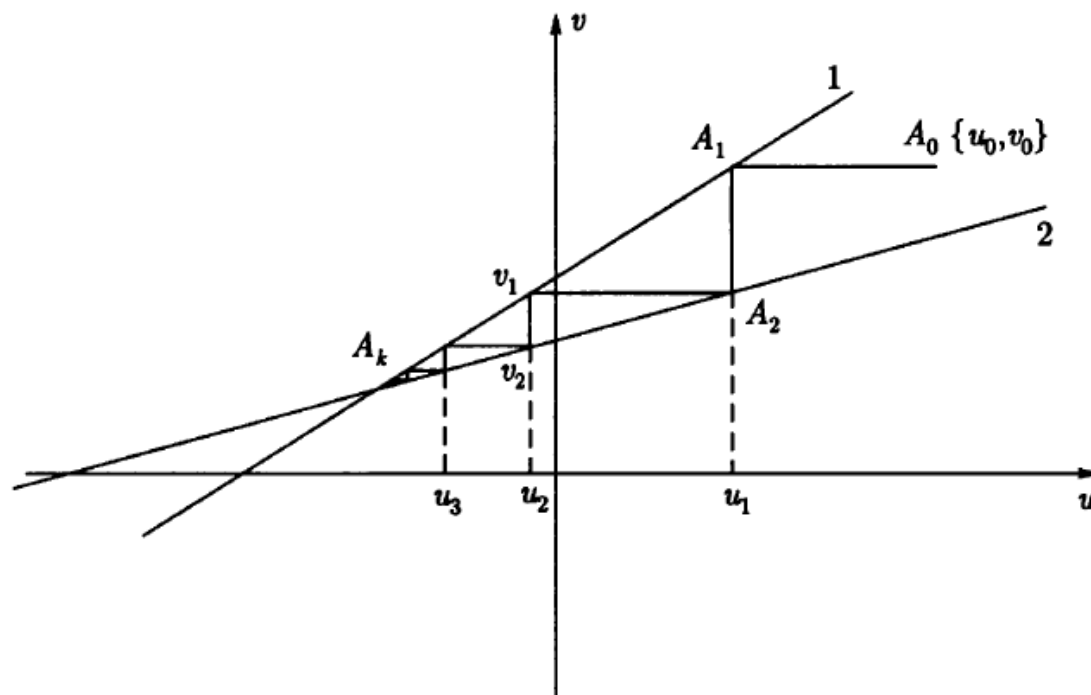
$$a_{21}u + a_{22}v = f_2,$$

$$a_{11} \neq 0, a_{22} \neq 0$$

Итерационный процесс Якоби записывается как

$$u_{k+1} = -\frac{a_{12}}{a_{11}}v_k + \frac{f_1}{a_{11}},$$

$$v_{k+1} = -\frac{a_{21}}{a_{22}}u_k + \frac{f_2}{a_{22}}.$$



Первое уравнение соответствует прямой 1, второе — прямой 2. Вычисление u_1 соответствует проведению отрезка, параллельного оси Ou и (при $v = v_0$) до пересечения с прямой 1; точка пересечения даст первое приближение u_1 . Вычислению v_1 соответствует проведение из точки A_1 прямой, параллельной оси Ov до пересечения с прямой 2 и т. д. до сходимости итераций к точке пересечения прямых 1 и 2 (A_k) с заданной точностью.

Пример. Методом простых итераций и методом Зейделя с точностью $\varepsilon = 0,003$ решить систему

$$\begin{cases} 5x_1 - 4x_2 - x_3 = -2; \\ 4x_1 + x_2 - 2x_3 = 8; \\ 3x_1 + x_2 - 5x_3 = 10. \end{cases}$$

▼ Вначале рассмотрим решение системы методом простых итераций и обеспечение достаточного условия сходимости этого метода. Система в исходном виде не может быть решена методом простых итераций, ибо простой перестановкой уравнений нельзя получить систему, в которой по диагонали стояли бы максимальные значения коэффициентов (здесь нет ни одного уравнения, в котором коэффициент при x_2 был бы максимален по модулю по сравнению с другими коэффициентами того же уравнения). Преобразуем систему. Вторым уравнением возьмем уравнение, полученное как разность первого и второго уравнений. Третье уравнение остается неизменным. В качестве первого уравнения возьмем второе уравнение исходной системы. Представим преобразованную таким образом систему (по диагонали будут коэффициенты с максимальными по модулю значениями) в итерационной форме:

$$\begin{cases} x_1 = 0 & - 0.25x_2 + 0.5x_3 + 2; \\ x_2 = 0.1x_1 + 0 & + 0.2x_3 + 2; \\ x_3 = 0.6x_1 + 0.2x_2 + 0 & - 2. \end{cases}$$

Проверяем условие сходимости:

$$\overline{C} = \begin{bmatrix} 0 & -0.25 & 0.5 \\ 0.2 & 0 & 0.2 \\ 0.6 & 0.2 & 0 \end{bmatrix}; \quad \|\overline{C}\| = \sqrt{\sum_{i=1}^3 \sum_{j=1}^3 c_{ij}^2} = \sqrt{0.79} < 1.$$

Достаточное условие сходимости для метода простых итераций выполнено. Результаты расчета по методу простой итерации и по методу Зейделя представлены в табл. 3.2 и 3.3, соответственно (в первых трех строках указаны приближения, в последней – решение).

Таблица 3.2. Значения приближённых решений, получаемых на последовательных итерациях при решении СЛАУ методом простых итераций

Итерации	x_1	x_2	x_3	$\max_i x_i^{(k+1)} - x_i^{(k)} $
0	2	2	-2	
1	0.500	2.000	-0.400	1.600
2	1.300	2.020	-1.300	0.900
.....
11	0.999	2.000	-0.999	0.003

Таблица 3.3. Значения приближённых решений, получаемых на последовательных итерациях при решении СЛАУ методом Зейделя

Итерации	x_1	x_2	x_3	$\max_i x_i^{(k+1)} - x_i^{(k)} $
0	2	2	-2	
1	0.500	1.700	-1.360	1.500
2	0.895	1.907	-1.082	0.395
.....
6	1.000	2.000	-1.000	0

Решение показывает большую скорость сходимости метода Зейделя. ▲

Пример 3. Решение систем линейных уравнений методом Зейделя.

Рассмотрим параллельно решение 3-х систем уравнений:

$$\begin{cases} 2x_1 + x_2 = 3 \\ x_1 - 2x_2 = 1 \end{cases}, \quad \begin{cases} x_1 + 2x_2 = 3 \\ 2x_1 - x_2 = 1 \end{cases}, \quad \begin{cases} 2x_1 - 0.5x_2 = 3 \\ 2x_1 + 0.5x_2 = 1 \end{cases}$$

Приведем системы к виду удобному для итераций:

$$\begin{cases} x_1^{(n+1)} = -0.5x_2^{(n)} + 1.5 \\ x_2^{(n+1)} = 0.5x_1^{(n+1)} - 0.5 \end{cases}, \quad \begin{cases} x_1^{(n+1)} = -2x_2^{(n)} + 3 \\ x_2^{(n+1)} = 2x_1^{(n+1)} - 1 \end{cases}, \quad \begin{cases} x_1^{(n+1)} = 0.25x_2^{(n)} + 1.5 \\ x_2^{(n+1)} = -4x_1^{(n+1)} + 2 \end{cases}$$

Заметим, что условие сходимости $\|B\| < 1$ выполнено только для первой системы. Вычислим 3 первых приближения к решению в каждом случае.

1-ая система. $x^{(0)} = \begin{pmatrix} 1.5 \\ -0.5 \end{pmatrix}, \quad x^{(1)} = \begin{pmatrix} 1.75 \\ 0.375 \end{pmatrix}, \quad x^{(2)} = \begin{pmatrix} 1.3125 \\ 0.1563 \end{pmatrix}, \quad x^{(3)} = \begin{pmatrix} 1.4219 \\ 0.2109 \end{pmatrix}$

Точное решение здесь $\underline{x}_1 = 1.4, \quad \underline{x}_2 = 0.2$. Итерационный процесс сходится.

$$x^{(0)} = \begin{pmatrix} 3 \\ -1 \end{pmatrix}, \quad x^{(1)} = \begin{pmatrix} 5 \\ 9 \end{pmatrix}, \quad x^{(2)} = \begin{pmatrix} -15 \\ -31 \end{pmatrix}, \quad x^{(3)} = \begin{pmatrix} 65 \\ 129 \end{pmatrix}.$$

2-ая система.

итерационный процесс разошелся.

Точное решение $x_1 = 1$, $x_2 = 0.2$.

$$x^{(0)} = \begin{pmatrix} 1.5 \\ 2 \end{pmatrix}, \quad x^{(1)} = \begin{pmatrix} 2 \\ -6 \end{pmatrix}, \quad x^{(2)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix}, \quad x^{(3)} = \begin{pmatrix} 0 \\ 2 \end{pmatrix} - \text{итерационный}$$

3-я система.

процесс заиклился.

Точное решение $x_1 = 1$, $x_1 = 2$.

Для геометрической интерпретации полученных результатов постройте чертеж.

$V = \text{diag}(A)$ - выделяется вектор-столбец,
состоящий из элементов главной диагонали
матрицы A

$D = \text{diag}(V)$ - создаётся матрица D с элементами
вектора V на главной диагонали

$LL = \text{tril}(A)$ – из матрицы A выделяется нижняя
треугольная матрица с главной диагональю

$L = LL - D$ - убирается главная диагональ из
нижней треугольной матрицы

$UU = \text{triu}(A)$ - из матрицы A выделяется верхняя
треугольная матрица с главной диагональю

ЗАДАНИЕ. №1 Про звездные войны

Знаменитые контрабандисты Хан Соло и Чуббака снова попали в передрагу! Имперцы засекли их с нелегальным грузом, а гиперпривод как назло сломался, для того чтобы сделать гиперпрыжок, им необходимо вычислить 4 параметра (X1-X4) и вбить их в компьютер. На основании координат прыжка Чуи смог вывести такую систему уравнений:

$$\begin{cases} 3x_1 + 3x_2 + 6x_3 + 3x_4 = 6 \\ 3x_1 + x_2 + 5x_3 + x_4 = 2 \\ 2x_1 + x_2 + 4x_3 + 2x_4 = 1 \\ x_1 + 3x_2 + 3x_3 + 2x_4 = 6 \end{cases}$$

Помоги друзьям вычислить параметры, пока их не настигла империя!

Перед тем как решить задачу необходимо определить детерминант матрицы коэффициентов, её ранг, норму и число обусловленности. Задать точность решения системы уравнений. Предлагается решить систему методами простой итерации, Зейделя и Якоби, а затем сравнить результаты. Перед применением разных методов нужно проверять выполнение условий их сходимости, а в ходе выполнения - фиксировать число требуемых итераций. По проделанной работе необходимо подготовить отчет с кодом и описанием методов, полученных результатов и выводами об эффективности их использования. Также решить задачу, используя оператор `linsolve(A,b)`.