Лекция 4.

Решение систем линейных уравнений методом простых итераций.

Если система имеет большую размерность (10^3 – 10^6 уравнений) или матрица системы ***разрежена***, более эффективны для решения непрямые итерационные методы. Разреженная матрица возникают в системе, где многие коэффициенты при неизвестных равны нулю.

Итерационные методы (методы последовательных приближений) состоят в том, что решение системы (1) находится как предел последовательных приближений $\bar{x}^{(n)}$ при $n \to \infty$, где n номер итерации. При использовании методов итерации обычно задается некоторое малое число $\varepsilon > 0$ и вычисления проводятся до тех пор, пока не будет выполнена оценка $\left\|\bar{x}^{(n)} - \bar{x}\right\| < \varepsilon$. К этим методам относятся метод Зейделя, Якоби, метод верхних релаксаций и т.д.

Рассмотрим один из простых методов – метод простой итерации.

Пусть дана система линейных уравнений, имеющая вид: $\begin{cases} a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n = b_2 \\ \qquad\qquad ... \\ a_{n}x_1 + a_{n2}x_2 + ... + a_{nn}x_n = b_n \end{cases}$ (1).

Эта система эквивалентна векторной (матричной) записи $A \cdot \vec{X} = \vec{B}$, где A – матрица системы,

$\vec{X} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$ - вектор – столбец неизвестных,

$\vec{B} = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix}$ - вектор – столбец свободных членов.

Данную систему можно преобразовать к виду $\vec{X} = C \cdot \vec{X} + \vec{F}$ (2), где C – некоторая матрица, $\vec{F}$ - вектор – столбец.

Обозначим $C \cdot \vec{X} + \vec{F} = \Phi(\vec{X})$, то есть система принимает форму $\vec{X} = \Phi(\vec{X})$. То есть, начиная с некоторого начального значения (нулевого приближения) получим последовательность.

$$\vec{X}^{(1)} = \Phi\left(\vec{X}^{(0)}\right)$$

$$\vec{X}^{(2)} = \Phi\left(\vec{X}^{(1)}\right)$$

…

$$\vec{X}^{(k+1)} = \Phi\left(\vec{X}^{(k)}\right).$$

Аналогичную последовательность мы получали при решении нелинейного уравнения методом простой итерации $x = \varphi(x)$.

Возникает только вопрос о том, как привести систему (1) к виду ***удобному для итераций*** (2). В общем случае это не простая задача, требующая специальных знаний, но в некоторых случаях можно поступить очень просто. Например, можно сделать такие преобразования:

$A \cdot \vec{X} = (A - I + I) \cdot \vec{X} = \vec{B}$, где I – единичная матрица.

$(A - I) \cdot \vec{X} + I \cdot \vec{X} = \vec{B}$

$\vec{X} = (I - A) \cdot \vec{X} + \vec{B}$

Самый простой способ следующий. Из первого уравнения системы (1) выразим неизвестное x_1:

$$x_1 = \frac{1}{a_{11}}\left(b_1 - a_{12}x_2 - a_{13}x_3 \ldots - a_{1n}x_n\right)$$

Из второго уравнения x_2:

$x_2 = \dfrac{1}{a_{22}}\left(b_2 - a_{21}x_1 - a_{23}x_3 \ldots - a_{2n}x_n\right)$ и так далее. В результате получим систему:

$$\begin{cases} x_1 = \qquad\qquad c_{12}x_2 + c_{12}x_3 + \ldots + c_{1n}x_n + f_1 \\ x_2 = c_{21}x_1 + \qquad\qquad c_{23}x_3 + \ldots + c_{2n}x_n + f_2 \\ \ldots \\ x_n = c_{n1}x_1 + c_{n2}x_2 + c_{n3}x_3 + \ldots + \quad \ldots \quad + f_n \end{cases} \quad (2).$$

На главной диагонали матрицы C стоят нули, а ненулевые элементы выражают по формулам:

$$c_{ij} = -\frac{a_{ij}}{a_{ii}},\; f_i = \frac{b_i}{a_{ii}} \;(i, j = 1, 2, \ldots n;\; i \neq j) \quad (3)$$

Для выполнения данного преобразования необходимо чтобы диагональные элементы исходной матрицы не были нулевыми. Если матрица преобразована к виду (2) по данным формулам (3), то метод итераций называют ***методом Якоби***.

Для системы (1) метод итерации сходится, если модули диагональных коэффициентов для каждого уравнения системы больше суммы модулей всех остальных коэффициентов (не считая свободных членов).

В качестве критерия окончания расчета (достижения заданной точности) можно использовать простое соотношение $\left|\vec{X}^{(n+1)} - \vec{X}^{(n)}\right| < \varepsilon$.

Пример. Решим систему уравнений с точностью $\varepsilon = 0{,}001$

$$\begin{cases} 6{,}25x_1 - x_2 + 0{,}5x_3 = 7{,}5 \\ -x_1 + 5x_2 + 2{,}12x_3 = -8{,}68 \\ 0{,}5x_1 + 2{,}12x_2 + 3{,}6x_3 = -0{,}24 \end{cases}$$

Перепишем систему пересчитав коэффициенты по формулам (3).

$$\begin{cases} x_1 = \qquad\qquad 0{,}16x_2 - 0{,}08x_3 + 1{,}2 \\ x_2 = 0{,}2x_1 - \qquad\qquad 0{,}424x_3 - 1{,}786 \\ x_3 = -0{,}1389x_1 - 0{,}58889x_2 \qquad - 0{,}0667 \end{cases}$$

Здесь $C = \begin{pmatrix} 0 & 0.16 & -0.08 \\ 0.2 & 0 & -0.424 \\ -0.1389 & -0.5889 & 0 \end{pmatrix}$, $F = \begin{pmatrix} 1.2 \\ -1.786 \\ -0.0667 \end{pmatrix}$.

За начальное приближение примем $\vec{X}^{(0)} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

Результаты итераций запишем в таблицу.

n	0	1	2	3	4
$x_1^{(n)}$	0	1,2	0,9276	0,902	0,8449
$x_2^{(n)}$	0	-1,736	-1,4677	-1,885	-1,8392
$x_3^{(n)}$	0	-0,0667	0,789	0,6688	0,9181
$\left\lvert \vec{X}^{(n+1)} - \vec{X}^{(n)} \right\rvert$		1,736	0,8557	0,4173	0,2493

n	…	12	13	14	15
$x_1^{(n)}$	…	0,8006	0,8003	0,8002	0,8001
$x_2^{(n)}$	…	-1,9985	-1,9993	-1,9995	-1,9998
$x_3^{(n)}$	…	0,9987	0,9990	0,9995	0,9997
$\left\lvert \vec{X}^{(n+1)} - \vec{X}^{(n)} \right\rvert$	…	0,0018	0,0008	0,0005	0,0003

После 15 итераций достигнута искомая точность, и можно записать решение. $x_1 = 0{,}8$; $x_2 = -2{,}0$; $x_3 = 1{,}0$.

Метод Зейделя.

Модификацией метода простых итераций Якоби можно считать метод Зейделя.

В методе Якоби на (k+1)-ой итерации значения x_i^{k+1} (i = 1, 2, …, n) вычисляются подстановкой в правую часть системы (2) вычисленных на предыдущей итерации значений x_i^k. В методе Зейделя при вычислении x_i^{k+1} используются значения x_1^{k+1} , x_2^{k+1} , x_{i-1}^{k+1}, уже найденные на (k+1)-ой итерации, а не x_1^k, x_2^k, …, x_{i-1}^k, как в методе Якоби, т.е. (k+1)-е приближение строится следующим образом:

$$\begin{cases} x_1^{k+1} = \qquad c_{12}x_2^k + c_{13}x_3^k + \ \dots \ + c_{1n}x_n^k + f_1 \\ x_2^{k+1} = c_{21}x_1^{k+1} + \qquad c_{23}x_3^k + \ \dots \ + c_{2n}x_n^k + f_2 \\ x_2^{k+1} = c_{31}x_1^{k+1} + c_{32}x_2^{k+1} + \qquad c_{34}x_3^k + \dots + c_{3n}x_n^k + f_3 \\ \dots \\ x_n^{k+1} = c_{n1}x_1^{k+1} + c_{n2}x_2^{k+1} + c_{n3}x_3^{k+1} + \dots + c_{nn-1}x_{n-1}^{k+1} + f_n \end{cases} \quad (4).$$

Это *расчетные формулы метода Зейделя.*

Запишем нижнюю и верхнюю треугольные матрицы:

$$C_1 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ c_{21} & 0 & 0 & \dots & 0 \\ c_{31} & c_{32} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ c_{n1} & c_{n2} & c_{n3} & \dots & 0 \end{pmatrix} \text{ и } C_2 = \begin{pmatrix} 0 & c_{12} & c_{13} & \dots & c_{1n} \\ 0 & 0 & c_{23} & \dots & c_{2n} \\ 0 & 0 & 0 & \dots & c_{3n} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & c_{nn} \end{pmatrix}$$

Матричная форма формул (4) имеет вид:

$$\vec{x}^{k+1} = C_1\vec{x}^{k+1} + C_2\vec{x}^{k+1} + \vec{F}$$

Сходимость метода Зейделя.

Достаточным условием сходимости метода Зейделя является выполнение неравенства: $max\left\lvert c_{ij} \right\rvert < 1, i, j = 1,2,...,n$. Это означает, что для сходимости метода Зейделя достаточно, чтобы максимальный по модулю элемент матрицы C =C_1+C_2 был меньше единицы.

Критерий окончания расчета можно взять такой же, как и в методе Якоби. Метод Зейделя, как правило, сходится быстрее, чем метод Якоби. Однако возможны ситуации, когда метод Якоби сходится, а метод Зейделя сходится медленнее или вообще расходится.

Пример.

Решим систему с точностью $\varepsilon = 0{,}001$ методом Зейделя.

$$\begin{cases} 3{,}2x_1 - 11{,}5x_2 + 3{,}8x_3 = 2{,}8 \\ 0{,}8x_1 + 1{,}3x_2 - 6{,}4x_3 = -6{,}5 \\ 2{,}4x_1 + 7{,}2x_2 - 1{,}2x_3 = 4{,}5 \end{cases}$$

Чтобы выполнить условия сходимости к первому уравнению прибавим третье, а второе и третье поменяем местами.

$$\begin{cases} 5{,}6x_1 - 4{,}3x_2 + 2{,}6x_3 = 7{,}3 \\ 2{,}4x_1 + 7{,}2x_2 - 1{,}2x_3 = 4{,}5 \\ 0{,}8x_1 + 1{,}3x_2 - 6{,}4x_3 = -6{,}5 \end{cases}$$

Выразим неизвестные:

$$\begin{cases} x_1 = \qquad\qquad 0{,}77x_2 - 0{,}46x_3 + 1{,}3 \\ x_2 = -0{,}333x_1 - \qquad\qquad 0{,}1667x_3 + 0{,}625 \\ x_3 = 0{,}125x_1 + 0{,}203x_2 \qquad + 1{,}015625 \end{cases}$$

Результаты итераций запишем в таблицу.

n	0	1	2	3	4
$x_1^{(n)}$	0	1,303571	0,884673	1,139681	1,06956
$x_2^{(n)}$	0	0,190476	0,532986	0,450852	0,476757
$x_3^{(n)}$	0	1,217262	1,234472	1,249664	1,246161
$\left\lvert \vec{X}^{(n+1)} - \vec{X}^{(n)} \right\rvert$		1,793685	0,948818	0,183948	0,186726

n	…	7	8	9	10
$x_1^{(n)}$	…	1,086559	1,085969	1,086147	1,086094
$x_2^{(n)}$	…	0,470637	0,470851	0,470786	0,470806
$x_3^{(n)}$	…	1,247043	1,247013	1,247022	1,247019
$\left\lvert \vec{X}^{(n+1)} - \vec{X}^{(n)} \right\rvert$	…	0,004575	0,001381	0,000417	0,000126

После 10 итераций достигнута искомая точность, и можно записать решение. $x_1 = 1{,}086$; $x_2 = 0{,}47$; $x_3 = 1{,}247$.

Методы численной интерполяции и аппроксимации.

Процесс научного познания часто приводит к необходимости обработки числовой информации. При обращении с данными научного и инженерного характера особенно важными являются такие вычислительные средства как ***интерполяция и приближение кривыми (аппроксимация).***

Некоторые типичные ситуации:

1. Функция f задана таблицей своих значений: $y_i = f(x_i)$. Требуется найти значение в точке $x \neq x_i$.

2. Функция f имеет громоздкое выражение и требует для вычисления значительных затрат времени.

3. Значения функции f найдены экспериментально с погрешностью, т.е. y_i^* вместо y_i.

Возникающие проблемы решают следующим образом: функцию $f(x)$ заменяют на $g(x)$, значения которой принимают за приближенные значения функции $f(x)$.

Линейная интерполяция.

Простейшим видом интерполяции является ***кусочно-линейная интерполяция***. Она состоит в том, что заданные точки (x_i , y_i), $i = 0, ..., n$ соединяются прямолинейными отрезками, и функция $f(x)$ приближается полученной ломаной.

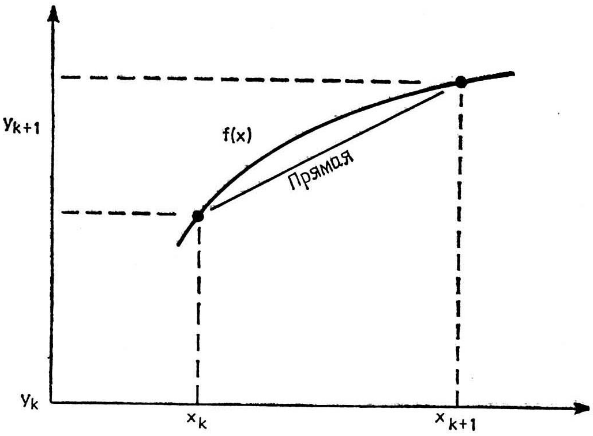
Уравнения каждого отрезка ломаной в общем случае разные. Для каждого из интервалов (x_{i-1}, x_i) в качестве уравнения интерполянты используется уравнения прямой, проходящей через две точки. В частности, для i-го интервала можно написать уравнение прямой, проходящей через точки (x_{i-1} , y_{i-1}) и (x_i ,y_i), в виде

$(y - y_{i-1})/(y_i - y_{i-1}) = (x - x_{i-1})/(x_i - x_{i-1})$

Отсюда

$y = a_i x + b_i \equiv g(x),\ x \in (x_{i-1}, x_i)$

$a_i = (y_i - y_{i-1})/(x_i - x_{i-1}),\ b_i = y_{i-1} - a_i x_{i-1},\ i = 1, ..., n.$



Следовательно, при использовании кусочно-линейной интерполяции сначала нужно определить номер i интервала, в который попадает значение аргумента x, затем подставить x и i в формулу и найти приближенное значение функции в точке x.

Обычно полагают, что аппроксимируя истинную кривую более сложной линией можно уточнить полученный результат. Например, можно найти многочлен n – ной степени $P_n(x)$, аппроксимирующий функцию $f(x)$ кривой, проходящей через все n + 1, заданные в таблице точки (x_i , y_i), где $i = 0, ..., n$. В этом случае многочлен должен удовлетворять условиям $P_n(x_i) = y_i$, для всех i.

Точки (x_i , y_i) называют ***узлами интерполирования***, а искомый многочлен – ***интерполяционным***.

Интерполяция многочленами.

Одна из форм записи интерполяционного многочлена - ***многочлен Лагранжа***.

Интерполяционный многочлен для этого метода запишем в виде.

$P_n(x) = y_0 b_0(x) + y_1 b_1(x) + \ldots + y_n b_n(x)$, где все $b_j(x)$ – многочлены степени n, коэффициенты которых можно найти с помощью n + 1 уравнений $P_n(x_i) = y_i$, $i = 0, ..., n$.

Подставим данные условия в интерполяционный многочлен и получим систему уравнений:

$y_0 b_0(x_0) + y_1 b_1(x_0) + \ldots + y_n b_n(x_0) = y_0$

$\ldots$

$y_0 b_0(x_n) + y_1 b_1(x_n) + \ldots + y_n b_n(x_n) = y_n$

Если значения $b_j(x_i)$ выбраны так, что

$b_j(x_i)=\begin{cases}1 \text{ при } i=j\\ 0 \text{ при } i\neq j\end{cases}$, то уравнения будут удовлетворены.

Это означает, что любой многочлен $b_j(x)$ равен нулю во всех точках x_i кроме x_j и его можно представить в виде:

$$b_j(x)=C_j(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)$$

Так как $b_j(x_j)=1$, то коэффициент C_j определяется выражением

$$C_j=1/(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)$$

Т.е многочлен

$b_j(x)=\dfrac{(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)}$ является требуемым многочленом степени. Подставим все в формулу для $P_n(x)$ и получим.

$$P_n(x)=\sum_{j=0}^{n}\frac{(x-x_0)(x-x_1)\cdots(x-x_{j-1})(x-x_{j+1})\cdots(x-x_n)}{(x_j-x_0)(x_j-x_1)\cdots(x_j-x_{j-1})(x_j-x_{j+1})\cdots(x_j-x_n)}y_j=L_n(x)$$

Это интерполяционный многочлен Лагранжа.

В инженерной практике наиболее часто используется интерполяция многочленами первой, второй и третьей степени (*линейная*, *квадратичная* и *кубическая интерполяции*).

При $n=1$ – линейная формула.

$L_1(x)=y_0\dfrac{x-x_1}{x_0-x_1}+y_1\dfrac{x-x_0}{x_1-x_0}$. (Совпадает с уже полученной формулой линейной интерполяции по двум узлам x_0, x_1.)

При $n=2$

$$L_2(x)=y_0\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)}+y_1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)}+y_2\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)}.$$

Существует и другой подход для построения интерполяционного многочлена— метод Ньютона (метод разделённых разностей). Но фактически формулы Лагранжа и Ньютона порождают один и тот же полином, разница только в алгоритме его построения.

Интерполирование сплайнами.

Можно повысить точность интерполяции функции путем повышения степени многочлена Лагранжа. Но это связано с повышением сложности вычисления и накоплением ошибок. Поэтому на практике для того, чтобы достаточно хорошо приблизить функцию, вместо построения интерполяционного многочлена высокой степени используют интерполирование кусочными многочленами.

Пусть отрезок $[a, b]$ точками $a=x_1<x_2<\ldots<x_n=b$ разбит на n частичных отрезков $[x_i, x_{i+1}]$, $i=1, \ldots, n-1$.

Сплайном называется функция, которая вместе с несколькими производными непрерывна на всем заданном отрезке $[a, b]$, а на каждом частичном отрезке $[x_i, x_{i+1}]$ является некоторым алгебраическим многочленом.

Термин «сплайн» происходит от английского слова «spline» (гибкая линейка, стержень) – названия приспособления, использовавшегося чертежниками для проведения гладких кривых через заданные точки.

Разность $m-p$ между степенью сплайна и наивысшим порядком непрерывной на отрезке $[a, b]$ производной называется *дефектом сплайна*.

Простейшим примером сплайна является кусочно-линейная функция, которая является сплайном первой степени с дефектом равным единице.

Наиболее употребительными на практике являются сплайны третьей степени, имеющие на $[a, b]$, по крайней мере, непрерывную первую производную. Они называются кубическими и обозначаются $S_3(x)$.

Такие сплайны на каждом из частичных отрезков $[x_i, x_{i+1}]$ совпадают с многочленом $S_3(x) = a_i + b_i x + c_i x^2 + d_i x^3$. В общем случае сплайн задается глобальным способом, т.е. с использованием всех узлов при любом их расположении. Но гораздо проще найти локально заданный сплайн, который на частичном отрезке имеет следующее выражение:

$$S_3(x) = \frac{(x_{i+1} - x)^2 (2(x - x_i) + h)}{h^3} y_i + \frac{(x - x_i)^2 (2(x_{i+1} - x) + h)}{h^3} y_{i+1} +$$

$$+ \frac{(x_{i+1} - x)^2 (x - x_i)}{h^2} m_i + \frac{(x - x_i)^2 (x - x_{i+1})}{h^2} m_{i+1},$$ где $m_i = \frac{y_{i+1} - y_{i-1}}{2h}$, $i = 1, 2, n\text{-}1$.

$$m_0 = \frac{4y_1 - y_2 - 3y_0}{2h};\ m_n = \frac{3y_n + y_{n-2} - 4y_{n-1}}{2h};\ h = \frac{b-a}{n}.$$

Если заданное значение x из первого $[x_0, x_1]$ или последнего $[x_{n-1}, x_n]$ отрезка, то вычисляются значения m_0 или m_n.

Пример. По таблице значений функции $y = f(x)$ найти значение $f(1{,}417)$.

i	0	1	2	3	4	5
x_i	1,2	1,3	1,4	1,5	1,6	1,7
y_i	1,5094	1,6994	1,9043	2,1293	2,3756	2,6456

Решение.

Заданное значение $x = 1{,}417$ находится на отрезке $[x_2; x_3] = [1{,}4; 1{,}5]$, $i = 2$, $[a, b] = [1{,}2; 1{,}7]$, $n = 5$, $h = 0{,}1$.

По формулам имеем: $m_2 = \frac{y_3 - y_1}{2h} = \frac{2{,}1293 - 1{,}6984}{2 \cdot 0{,}1} = 2{,}1545$

$$m_3 = \frac{y_4 - y_2}{2h} = \frac{2{,}3756 - 1{,}9043}{2 \cdot 0{,}1} = 2{,}3565$$

Теперь подставим все в формулу сплайна.

$$f(1{,}417) \approx S_3(1{,}417) = \frac{(1{,}5 - 1{,}417)^2 (2(1{,}417 - 1{,}4) + 0{,}1)}{0{,}1^3} 1{,}9043 +$$

$$+ \frac{(1{,}417 - 1{,}4)^2 (2(1{,}5 - 1{,}417) + 0{,}1)}{0{,}1^3} 2{,}1293 +$$

$$+ \frac{(1{,}5 - 1{,}417)^2 (1{,}417 - 1{,}4)}{0{,}1^2} 2{,}1545 + \frac{(1{,}417 - 1{,}4)^2 (1{,}417 - 1{,}5)}{0{,}1^2} 2{,}3565 = 1{,}9415$$