

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение высшего профессионального
образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ
УНИВЕРСИТЕТ»**

Численные методы решения прикладных задач

Учебно-методические указания к
выполнению лабораторных работ по курсу Информатика

Томск 2011 г

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Решение алгебраических и трансцендентных уравнений	5
1.1 Общие положения	5
1.1 Метод половинного деления	5
1.2 Метод Ньютона	7
1.3 Метод секущих	9
2 Решение систем линейных уравнений	10
2.1 Общие положения	10
2.2 Метод Гаусса	10
2.2.1 Пример	12
2.3 Метод итераций	14
2.4 Метод Зейделя	16
3 Интерполяция, полином Лагранжа	18
3.1 Общие положения	18
3.2 Интерполяционный полином	18
3.2.1 Пример	19
3.3 Интерполяционная формула Лагранжа	19
4 Численное интегрирование	21
4.1 Общие положения	21
4.2 Метод прямоугольников	22
4.3 Метод трапеции	24
4.4 Метод Симпсона	25
ЛИТЕРАТУРА	27

ВВЕДЕНИЕ

Часто возникает необходимость, как в самой математике, так и ее приложениях в разнообразных областях получать решения математических задач в числовой форме. (Для представления решения в графическом виде также требуется предварительно вычислять его значения.) При этом для многих задач известно только о существовании решения, но не существует конечной формулы, представляющей ее решение. Даже при наличии такой формулы ее использование для получения отдельных значений решения может оказаться неэффективным. Наконец, всегда существует необходимость решать и такие математические задачи, для которых строгие доказательства существования решения на данный момент отсутствуют.

Во всех этих случаях используются методы приближенного, в первую очередь численного решения. Методы численного решения математических задач всегда составляли неотъемлемую часть математики и неизменно входили в содержание естественно-математического и инженерного образования. Как самостоятельная математическая дисциплина вычислительная математика оформилась в начале 20-го века. К этому времени в основном были разработаны разнообразные, достаточно эффективные и надежные алгоритмы приближенного решения широкого круга математических задач, включающего стандартный набор задач из алгебры, математического анализа и дифференциальных уравнений.

Прогресс в развитии численных методов способствовал постоянному расширению сферы применения математики в других научных дисциплинах и прикладных разработках, откуда в свою очередь поступали запросы на решение новых проблем, стимулируя дальнейшее развитие вычислительной математики. Метод математического моделирования, основанный на построении и исследовании математических моделей различных объектов, процессов и явлений и получении информации о них из решения связанных с этими моделями математических задач, стал одним из основных способов исследования в так называемых точных науках.

Современной формой метода математического моделирования, базирующейся на мощной вычислительной базе в виде ЭВМ и программного обеспечения, реализующего алгоритмы численного решения, является вычислительный эксперимент, рассматриваемый как новый теоретический метод исследования различных явлений и процессов. Этот теоретический метод включает существенные черты методологии экспериментального

исследования, но эксперименты выполняются не над реальным объектом, а над его математической моделью, и экспериментальной установкой является ЭВМ.

Технологическая цепочка вычислительного эксперимента включает в себя следующие этапы:

1. Построение математической модели исследуемого объекта (сюда же относится и анализ модели, выяснение корректности поставленной математической задачи).
2. Построение вычислительного алгоритма - метода приближенного решения поставленной задачи и его обоснование.
3. Программирование алгоритма на ЭВМ и его тестирование.
4. Проведение серии расчетов с варьированием определяющих параметров исходной задачи и алгоритма.
5. Анализ полученных результатов.

Каждый из этих этапов допускает возврат к любому из предыдущих этапов с целью его уточнения и корректировки. В данном курсе рассматриваются вопросы, связанные со вторым этапом вычислительного эксперимента. Во многих случаях вычислительный алгоритм решения сложной задачи строится из набора базовых компонент, представляющих собой алгоритмы решения некоторых стандартных математических задач. Изучение численных методов решения этих задач - необходимый элемент овладения современной технологией математического моделирования.

При этом идея модели лежит в основе того, что можно назвать методом вычислительной математики. Как правило, алгоритмы приближенного решения базируются на том, что исходная математическая задача заменяется (аппроксимируется) некоторой более простой или чаще последовательностью более простых задач. Решение этих более простых задач трактуется как приближенное решение задачи исходной. Т.е. фактически используется некоторая модель исходной задачи.

1 Решение алгебраических и трансцендентных уравнений

1.1 Общие положения

Во многих научных и инженерных задачах возникает необходимость решения уравнений вида:

$$F(x, p_1, p_2, \dots, p_n) = 0, \quad (1)$$

где F - заданная функция, x - неизвестная величина, $p_1, p_2, \dots, p_n$ - параметры задачи.

Решениями или корнями уравнения (1) называются такие значения x , которые при подстановке в уравнение обращают его в тождество.

Только для простейших уравнений удается найти решение в аналитическом виде, т.е. записать формулу, выражающую искомую величину x в явном виде. В большинстве же случаев приходится решать уравнение вида (1) численными методами. Хотя иногда, даже при наличии аналитического решения, имеющего сложный вид, бывает проще провести численное решение по известному алгоритму, чем программировать громоздкую аналитическую формулу.

Численное решение уравнения (1) обычно проводят в два этапа.

На первом этапе необходимо определить интервал изменения переменной x , где расположен один корень или, что означает то же самое, определить достаточно хорошее приближение окрестности этой точки.

На втором этапе тем или иным численным методом определяется величина x , соответствующая корню уравнения (1) с заданной погрешностью.

Для решения второй задачи существуют многочисленные методы, из которых мы рассмотрим лишь три:

- метод половинного деления;
- метод Ньютона;
- метод секущих.

1.1 Метод половинного деления

Для применения метода половинного деления необходимо установить окрестность или отрезок $[a, b]$, на котором расположен один из корней уравнения, который необходимо уточнить с погрешностью E (рисунок 1).

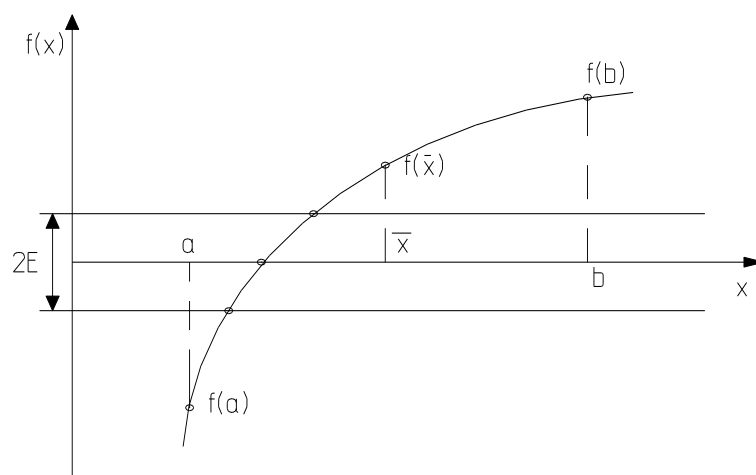


Рисунок 1 – Метод половинного деления

Пусть дано уравнение $f(x) = 0$, где $f(x)$ непрерывна на отрезке $[a, b]$ и $f(a) \cdot f(b) < 0$.

Метод половинного деления, или дихотомии, заключается в следующем. Для нахождения корня уравнения, принадлежащего отрезку $[a, b]$, делим отрезок пополам, т.е. выбираем начальное приближение, равное:

$$\bar{x} = \frac{(a+b)}{2}$$

и вычисляем значение функции $f(\bar{x})$. Если $f(\bar{x}) = 0$, то $\bar{x}$ является корнем уравнения. Если $f(\bar{x}) \neq 0$, то выбираем, одну из двух частей отрезка $[a, \bar{x}]$ или $[\bar{x}, b]$ для дальнейшего уточнения корня. Естественно, что корень будет находиться в той половине отрезка, на концах которого функция $f(x)$ имеет разные знаки, а именно проверяем условие: $f(a) \cdot f(\bar{x}) < 0$. На рисунке 1 это будет отрезок $[a, \bar{x}]$, т. е. для очередного шага уточнения точку b перемещаем в середину отрезка $\bar{x}$ ($b = \bar{x}$) и продолжаем процесс деления как с первоначальным отрезком $[a, b]$.

Итерационный (повторяющийся) процесс деления будет продолжаться до тех пор, пока не будет выполнено условие:

$$|b - a| \leq E.$$

За приближенное решение принимается средняя точка последнего промежутка.

Таким образом, для реализации метода дихотомии необходимо:

1. Задать в явном виде уравнение $f(x)$, корни которого необходимо определить.
2. Определить начальный интервал $[a, b]$, внутри которого лежит корень.
3. Задать точность нахождения корня уравнения $f(x)$.

4. Реализовать в программе итерационную процедуру, описанную выше.

1.2 Метод Ньютона

Предположим, что у нас определено начальное приближение x_0 к одному из корней уравнения (1). Тогда в точке x_0 можно вычислить левую часть решаемого уравнения $f(x_0)$.

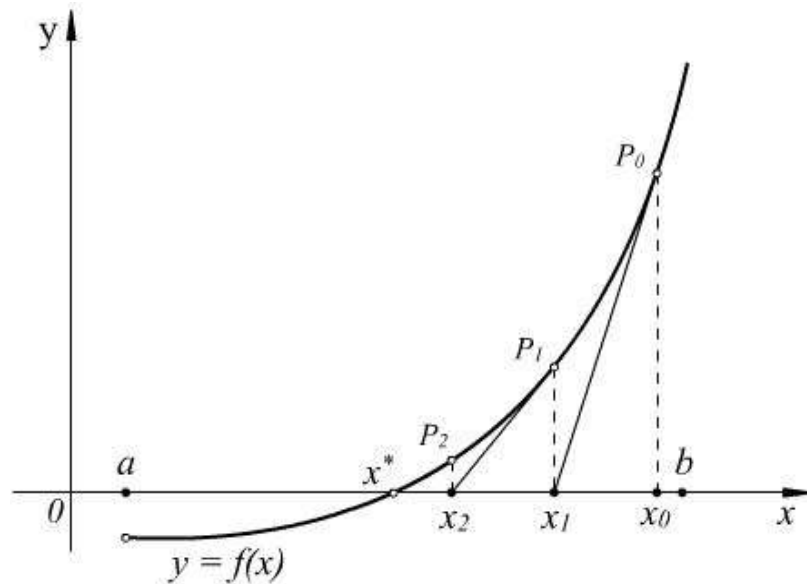


Рисунок 2 – Метод Ньютона

Рассмотрение метода Ньютона начнем с его геометрического представления (рисунок 2). Возьмем точку x_0 отрезка $[a, b]$ и проведем в точке P_0 с координатами $(x_0, f(x_0))$ касательную к кривой $y = f(x)$ до пересечения с осью Ox . Получим значение x_1 , в котором касательная пересекает ось Ox . Угловым коэффициентом касательной равен значению производной от функции $f(x)$ в точке касания. Следовательно, уравнение касательной, проходящей через точку с координатами $(x_0, f(x_0))$ имеет вид:

$$y = f(x_0) + f'(x_0) \cdot (x - x_0).$$

Полагая $y=0$, находим точку пересечения касательной с осью Ox , которую обозначим через x_1 :

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Абсциссу x_1 точки пересечения можно взять в качестве приближенного значения корня. Проведя касательную через новую точку с координатами $(x_1, f(x_1))$ и найдя точку ее пересечения с осью Ox , получим второе приближения корня x_2 . Аналогично определяются последующие приближения.

Следующие приближения находим соответственно по формулам:

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

.....

$$x_n = x_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$$

В общем случае для k -го шага итерационного процесса последнее соотношение принимает вид:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

(2)

Из формулы

(2)

(2) вытекает необходимость вычисления значения производной функции $f(x)$ в каждой точке. Процесс нахождения корня может считаться законченным, когда модуль отношения значения функции в точке x_k к ее производной меньше заданной величины погрешности

$\left| \frac{f(x_k)}{f'(x_k)} \right| < E$, т.е. когда выполняется следующее условие:

$$|x_{k+1} - x_k| < E \quad (3)$$

Таким образом, для реализации метода Ньютона необходимо:

1. Задать в явном виде уравнение $f(x)$, корни которого необходимо определить.
2. Определить первую производную функции $f(x)$ в аналитическом виде.
3. Определить начальное приближение x_0 , обеспечивающее быструю сходимость метода.
4. Задать точность нахождения корня уравнения $f(x)$.
5. Реализовать в программе итерационную процедуру, реализующую формулу
6. (2).

1.3 Метод секущих

Далеко не всегда бывает удобно находить аналитическое выражение для производной функции, в таком случае можно использовать метод секущих.

Для начала итерационного процесса необходимо задать два начальных приближения x_0 и x_1 .

Если x_0 и x_1 расположены достаточно близко друг к другу, то производную $f'(x_1)$ можно заменить ее приближенным значением в виде отношения приращения функции равного $f(x_1) - f(x_0)$ к отношению приращения аргумента равного $(x_1 - x_0)$:

$$f'(x_1) = \frac{f(x_1) - f(x_0)}{x_1 - x_0} \quad (4)$$

Таким образом, формула метода секущих может быть получена из формулы Ньютона (2) заменой производной выражением (4) и записана в виде:

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})} f(x_k) \quad (5)$$

Однако следует помнить, что при этом нет необходимости, чтобы значения функции $f(x_0)$ и $f(x_1)$ обязательно имели разный знак, как в методе половинного деления.

Процесс нахождения корня при использовании метода секущих можно считать законченным, когда выполняется следующее условие:

$$|x_{k+1} - x_k| < E \quad (6)$$

Метод секущих несколько уступает методу Ньютона в скорости сходимости, однако не требует вычислений производной левой части уравнения.

Таким образом, для реализации метода секущих необходимо:

1. Задать в явном виде уравнение $f(x)$, корни которого необходимо определить.
2. Определить начальные приближения x_0 и x_1 , обеспечивающие быструю сходимость метода.
3. Задать точность нахождения корня уравнения $f(x)$.
4. Реализовать в программе итерационную процедуру, реализующую формулу (2.5).

Результатом проведения лабораторной работы является программа, реализующая один из описанных методов с решением контрольного примера согласно, полученного индивидуального задания.

2 Решение систем линейных уравнений

2.1 Общие положения

При решении большого класса прикладных задач возникает необходимость в нахождении корней СЛАУ. Методы решения СЛАУ можно разделить на два больших класса: точные и итерационные.

Точные методы решения, например метод Гаусса, дают, вообще говоря, точное значение корней СЛАУ, при этом при корректном составлении программы точность определяется только погрешностями, связанными с округлением и представлением чисел в ЭВМ.

Итерационные методы решения СЛАУ характеризуется тем, что точное решение системы они могут, вообще говоря, давать лишь как предел некоторой бесконечной последовательности векторов. Исходное приближение при этом разыскивается каким-либо другим способом или задается произвольно. При выполнении определенных требований можно получить достаточно быстро сходящийся к решению итерационный процесс. К этому классу методов относятся: метод итераций и метод Зейделя.

2.2 Метод Гаусса

Рассмотрим задачу решения системы уравнений вида:

$$\begin{array}{ccccccc} a_{11}x_1 + & \dots & + a_{1n}x_n & = & b_1 \\ a_{21}x_1 + & \dots & + a_{2n}x_n & = & b_2 \\ \dots & \dots & \dots & \dots & \\ a_{n1}x_1 + & \dots & + a_{nn}x_n & = & b_n \end{array} \quad (7)$$

Известно, что система (7) имеет единственное решение, если ее матрица невырожденная (т. е. определитель матрицы отличен от нуля). В случае вырожденности матрицы система может иметь бесконечное число решений (если ранг матрицы и ранг расширенной матрицы, полученной добавлением к столбца свободных членов равны) или не иметь решений вовсе (если ранг матрицы и расширенной матрицы не совпадают).

Систему (3.1) можно записать в матрично-векторной форме $A X = B$,
где A - матрица коэффициентов системы, содержащая n строк и n столбцов;

B - заданный вектор правых частей;

X - искомый вектор.

Метод Гаусса основан на известном из обычного школьного курса алгебры методе исключений. Комбинируя каким-либо образом уравнения системы, добиваются того, что во всех уравнениях, кроме одного, будет исключено одно из неизвестных. Затем исключают другое неизвестное, третье и т.д.

Рассмотрим систему уравнений размера $n \times n$. Алгоритм гауссова исключения состоит из нескольких шагов. Если система записана в виде (7), то первый шаг состоит в исключении x_1 из последних $n-1$ уравнений. Это достигается вычитанием из второго уравнения первого, умноженного на a_{21}/a_{11} , из третьего уравнения первого, умноженного на a_{31}/a_{11} , и т.д. Этот процесс приводит к преобразованной системе уравнений:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{22}^{(1)}x_2 + \dots + a_{2n}^{(1)}x_n = b_2^{(1)} \\ \dots \dots \dots \\ a_{nn}^{(1)}x_n = b_n^{(1)} \end{cases} \quad (8)$$

где

$$a_{ij}^{(1)} = a_{ij} - a_{1j} \frac{a_{i1}}{a_{11}}, \quad b_i^{(1)} = b_i - b_1 \frac{a_{i1}}{a_{11}}, \quad i, j=2, \dots, n.$$

Применяя теперь тот же самый процесс к последним $n-1$ уравнениям системы (8), исключаем x_2 из последних $n-2$ уравнений и т.д., пока вся система не приведет к треугольной форме:

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ & & \dots & \\ 0 & \dots & a_{nn}^{(n-1)} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix} = \begin{bmatrix} b_1 \\ b_2^{(1)} \\ \dots \\ b_n^{(n-1)} \end{bmatrix}, \quad (9)$$

где верхние индексы, вообще говоря, указывают, сколько раз изменялись соответствующие коэффициенты. Этим завершается фаза прямого исключения (или приведением к треугольной форме) алгоритма гауссова исключения. Решение треугольной системы (9) теперь легко получается на фазе обратной подстановки, в ходе которой уравнения системы (9) решаются в обратном порядке:

[illegible]

При этом все диагональные коэффициенты должны быть отличны от нуля.

2.2.1 Пример

Существует большое количество модификаций вычислительных схем, реализующих метод Гаусса. В качестве примера рассмотрим компактную схему Гаусса. Для примера выбрана СЛАУ 3-го порядка.

$$4 \cdot x_1 - 9 \cdot x_2 + 2 \cdot x_3 = 2$$

$$2 \cdot x_1 - 4 \cdot x_2 + 4 \cdot x_3 = 3$$

$$1 \cdot x_1 + 2 \cdot x_2 + 2 \cdot x_3 = 1,$$

которая в матричной форме записывается в виде:

$$\begin{bmatrix} 4 & -9 & 2 \\ 2 & -4 & 4 \\ -1 & 2 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 3 \\ 1 \end{bmatrix} \quad (11)$$

Первый основной шаг гауссова исключения состоит в исключении первой переменной x_1 из второго и третьего уравнений. Если из второго уравнения системы вычесть первое, умноженное на 0.5, и из третьего уравнения вычесть первое, умноженное на -0.25 , то получим эквивалентную систему уравнений:

$$\begin{bmatrix} 4 & -9 & 2 \\ 0 & 0.5 & 3 \\ 0 & -0.25 & 2.5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 1.5 \end{bmatrix} \quad (12)$$

Второй основной шаг состоит в исключении x_2 из третьего уравнения. Это может быть сделано вычитанием из третьего уравнения второго, умноженного на -0.5 , что приводит к системе вида:

$$\begin{bmatrix} 4 & -9 & 2 \\ 0 & 0.5 & 3 \\ 0 & 0 & 4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 2.5 \end{bmatrix} \quad (13)$$

Прделанные операции называются *элементарными преобразованиями строк*. К этому моменту завершается первая часть алгоритма гауссова исключения, которую обычно называют *прямым исключением* или *приведением к треугольной форме*. Эта часть завершается тогда, когда все элементы последней строки системы, кроме крайне правого, обращаются в нуль.

Вторая часть алгоритма заключается в решении полученной *верхней треугольной системы*. Это легко осуществляется с помощью процесса *обратной подстановки*. Последнее уравнение системы

(13) имеет вид $4x_3=2.5$. Следовательно, $x_3=0.625$. Подставляя теперь это значение во второе уравнение: $0.5x_2+3\cdot 0.625=2$.

Отсюда $x_2=0.25$. Подстановка этих значений x_2 и x_3 в первое уравнение дает $4x_1-(9)(0.25)+(2)(0.625)=2$, или $x_1=0.75$. Чтобы проверить найденное решение, выполним умножение

$$\begin{bmatrix} 4 & -9 & 2 \\ 2 & -4 & 4 \\ -1 & 2 & 2 \end{bmatrix} \begin{bmatrix} 0.75 \\ 0.25 \\ 0.625 \end{bmatrix},$$

результат, которого совпадает с правой частью (13).

Процесс гауссова исключения можно очень компактно записать в виде алгоритма.

Прямое исключение

для $k=1, \dots, n-1$,

для $i=k+1, \dots, n$:

$$l_{ik} \leftarrow a_{ik} / a_{kk} ;$$

для $j=k, \dots, n$:

$$a_{ij} \leftarrow a_{ij} - l_{ik} a_{kj}$$

$$b_i \leftarrow b_i - l_{ik} b_k$$

Обратная подстановка

для $k=n, n-1, \dots, 1$:

$$x_k \leftarrow (b_k - \sum_{j=k+1}^n a_{kj} x_j) / a_{kk}$$

При составлении программы для ЭВМ, реализующей этот алгоритм, следует обратить внимание на то, что последовательно преобразуемые в ходе этого процесса элементы $a_{ij}^{(k)}$

На это указывает пятая строка алгоритма. Если это будет сделано, то исходная матрица, разумеется, будет испорчена.

При разработке алгоритма, реализующего метод **Гаусса**, на первом этапе рекомендуется преобразовать исходную матрицу к виду, когда на главной диагонали выстраиваются максимальные по абсолютной величине коэффициенты. При этом если хотя бы одно значение коэффициента, стоящего на главной диагонали, равно нулю, применять метод **Гаусса нельзя**.

Способ **итераций** дает возможность получить последовательность приближенных значений, сходящихся к точному решению системы.

$$\left\{ \begin{array}{l} a_{1,1} \cdot x_1 + a_{1,2} \cdot x_2 + \dots + a_{1,n} \cdot x_n + a_{1,n+1} = 0 \\ a_{2,1} \cdot x_1 + a_{2,2} \cdot x_2 + \dots + a_{2,n} \cdot x_n + a_{2,n+1} = 0 \\ \dots\dots\dots \\ a_{n,1} \cdot x_1 + a_{n,2} \cdot x_2 + \dots + a_{n,n} \cdot x_n + a_{n,n+1} = 0 \end{array} \right.$$

Разрешим первое уравнение относительно x_1 , второе относительно x_2 и т.д. Тогда систему

[illegible]
$$\text{где } b_{i,k} = -\frac{a_{i,k}}{a_{i,i}} \quad (i=1,\dots,n, \quad k=1,\dots, n+1)$$

(15) являются функциями переменных $x_1, x_2, \dots, x_n$. Обозначив правы части уравнений через $L_i(x_1, x_2, \dots, x_n)$, систему

14

$$\begin{aligned} x_1 &= L_1(x_1, x_2, \dots, x_n) \\ x_2 &= L_2(x_1, x_2, \dots, x_n) \\ &\vdots \\ x_n &= L_n(x_1, x_2, \dots, x_n) \end{aligned} \quad (16)$$

Итерации начинаются с задания начального приближенного решения $x_1^0, x_2^0, \dots, x_n^0$, которое может быть получено из физических или других разумных соображений. Чем ближе исходное приближение к решению, тем меньше итераций необходимо для его получения.

Для заданных начальных приближений $x_1^0, x_2^0, \dots, x_n^0$ после подстановки этих значений в правые части системы (16) получим первые приближения

$$\begin{aligned} x_1^1 &= L_1(x_1^0, x_2^0, \dots, x_n^0) \\ x_2^1 &= L_2(x_1^0, x_2^0, \dots, x_n^0) \\ &\dots\dots\dots \\ x_n^1 &= L_n(x_1^0, x_2^0, \dots, x_n^0) \end{aligned} \tag{17}$$

Полученные первые приближения могут быть таким же образом использованы для получения 2-х, 3-их и т.д., так что для любого m можно получить m -ое приближение $x^m_1, x^m_2, \dots, x^m_n$.

Итерации проводятся до получения решения с требуемой точностью, которая не должна превышать заданной погрешности вычислений, т.е. окончание итерационного процесса происходит при выполнении следующего неравенства:

$$\left| x_i^m - x_i^{m-1} \right| < E, \text{ npu } i=1, 2, \dots, n,$$

где E – заданная точность решения.

Естественно возникает вопрос об условиях, выполнение которых обеспечивает сходимость полученных приближений к истинному решению системы. Достаточное условие сходимости, т.е. возможности решения СЛАУ методом **итераций**, формулируется следующим образом.

Для того чтобы итерационный процесс сходился, достаточно, чтобы в любой строке сумма отношений коэффициентов системы к диагональным коэффициентам, взятым из той же строки была строго меньше единицы.

Математически это определение может быть выражено следующим образом:

$$\sum_{\substack{j=1 \\ j \neq i}}^n \frac{|a_{i,j}|}{|a_{i,i}|} < 1$$

Или, что то же самое,

$$\sum_{\substack{j=1 \\ i \neq j}}^n |a_{i,j}| < |a_{i,i}|,$$

т.е. можно сказать, что в любой строке исходной матрицы на главной диагонали должен находиться коэффициент, по абсолютному значению превосходящий сумму модулей остальных коэффициентов.

На первом этапе решения СЛАУ система приводится к виду (16), после чего происходит проверка условия сходимости итерационного процесса к решению системы. Для этого необходимо выбрать максимальные значения коэффициентов $a_{i,i}$ и провести проверку условия на сходимость итерационного процесса. После этого задаются начальные приближения, обычно для этого используется столбец свободных членов, и проводится расчет по формуле (17), которую можно представить в виде

$$x_i^m = \sum_{\substack{j=1 \\ j \neq i}}^n b_{i,j} \cdot x_j^{m-1} + b_{i,n+1},$$

до достижения окончательного решения. Здесь используется два вектора переменных: с предыдущими значениями X_0 и с последующими значениями X_1 . В конце каждой итерации производится переприсваивание значений из последующих в предыдущие.

2.4 Метод Зейделя

Метод **Зейделя** является модификацией метода **итераций**. СЛАУ задается в виде (14) и приводится к виду (15). Отличие от метода **итераций** заключается в вычислительной процедуре нахождения приближения на $i+1$ итерации. В отличие от метода простых **итераций**, где для отыскания $i+1$ приближения используется i -ое приближение неизвестных x_j^i , в методе **Зейделя** используются уже вычисленные $i+1$ значения x . Рекуррентные соотношения используемые в методе **Зейделя** представляются следующим образом:

$$\begin{aligned} x_1^{i+1} &= L_1(x_1^i, x_2^i, \dots, x_n^i) \\ x_2^{i+1} &= L_2(x_1^{i+1}, x_2^i, \dots, x_n^i) \\ &\dots\dots\dots \\ x_n^{i+1} &= L_n(x_1^{i+1}, x_2^{i+1}, \dots, x_n^i) \end{aligned} \tag{18}$$

Условия сходимости метода **Зейделя** может быть сформулировано следующим образом:

Для того чтобы итерационный процесс сходился, достаточно, чтобы сумма абсолютных значений элементов каждой строки (исключая диагональный) была меньше абсолютного значения диагонального элемента соответствующей строки.

Математически это определение может быть выражено следующим образом

$$|a_{i,i}| > \sum_{\substack{j=1, \\ i \neq j}}^n |a_{i,j}|$$

На первом этапе решения СЛАУ система приводится к виду (15), после чего происходит проверка условия сходимости итерационного процесса к решению системы. Для этого необходимо выбрать максимальные значения коэффициентов $a_{i,i}$ и провести проверку условия на сходимость итерационного процесса. После этого задаются начальные приближения, обычно для этого используется столбец свободных членов, и проводится расчет по формуле (18) до достижения окончательного решения.

3 Интерполяция, полином Лагранжа

3.1 Общие положения

В вычислительной практике часто приходится иметь дело с функциями $f(x)$, заданными таблицами их значений для некоторого конечного множества значений x : $f(x_0), f(x_1), \dots, f(x_n)$.

В процессе же решения задачи необходимо использовать значения $f(x)$ для промежуточных значений аргумента. В этом случае строят функцию $\Phi(x)$, достаточно простую для вычислений, которая в заданных точках $x_0, x_1, \dots, x_n$, называемых узлами интерполяции, принимает значения $f(x_0), f(x_1), \dots, f(x_n)$, а в остальных точках отрезка (x_0, x_n) , принадлежащего области определения $f(x)$, приближенно представляет функцию $f(x)$ с той или иной степенью точности.

При решении задачи в этом случае вместо функции $f(x)$ оперируют с функцией $\Phi(x)$. Задача построения такой функции $\Phi(x)$ называется задачей интерполирования. Чаще всего интерполирующую функцию $\Phi(x)$ отыскивают в виде алгебраического полинома.

3.2 Интерполяционный полином

Для каждой функции $f(x)$, определенной на $[a, b]$, и любого набора узлов $x_0, x_1, \dots, x_n$ ($x_i \in [a, b]$, $x_i \neq x_j$ при $i \neq j$) среди алгебраических многочленов степени не выше n существует единственный интерполяционный многочлен $\Phi(x)$, который может быть записан в форме:

$$\Phi(x) = \sum_{i=0}^n f(x_i) \Psi_i(x), \quad (19)$$

где $\Psi_i(x)$ - многочлен n -ой степени, обладающий следующим свойством:

$$\Psi_i(x) = \begin{cases} 1, & \text{при } x = x_i \\ 0 & \text{при } x = x_j \end{cases} \quad \text{для } (i, j) = 0, 1, 2, \dots, n, \quad i \neq j \quad (20)$$

Для интерполяционного полинома многочлен $\Psi_i(x)$ имеет вид:

$$\Psi_i = \frac{(x - x_0)(x - x_1) \dots (x - x_{i-1})(x - x_{i+1}) \dots (x - x_n)}{(x_i - x_0) \dots (x_i - x_{i-1})(x_i - x_{i+1}) \dots (x_i - x_n)} \quad (21)$$

Многочлен (19) и решает задачу интерполирования и называется интерполяционным полиномом Лагранжа.

3.2.1 Пример

В качестве примера рассмотрим функцию вида $f(x) = x^2$ на интервале 1,4 заданную табличным способом.

X	1	2	3	4
F(x)	1	4	9	16

Необходимо определить значение функции в точке $x = 2.5$. Воспользуемся для этого полином Лагранжа. Исходя из формул ((19) и (21)) запишем этот полином в явном виде:

$$\begin{aligned} \Phi(x) = & f(x_0) \frac{(x-x_1)(x-x_2)(x-x_3)}{(x_0-x_1)(x_0-x_2)(x_0-x_3)} + f(x_1) \frac{(x-x_0)(x-x_2)(x-x_3)}{(x_1-x_0)(x_1-x_2)(x_1-x_3)} + \\ & + f(x_2) \frac{(x-x_0)(x-x_1)(x-x_3)}{(x_2-x_0)(x_2-x_1)(x_2-x_3)} + f(x_3) \frac{(x-x_0)(x-x_1)(x-x_2)}{(x_3-x_0)(x_3-x_1)(x_3-x_2)} \end{aligned} \quad (22)$$

Тогда подставляя в формулу (22) исходные значения из нашей таблицы получим

$$\begin{aligned} \Phi(2.5) = & 1 \frac{(2.5-2)(2.5-3)(2.5-4)}{(1-2)(1-3)(1-4)} + 4 \frac{(2.5-1)(2.5-3)(2.5-4)}{(2-1)(2-3)(2-4)} + 9 \frac{(2.5-1)(2.5-2)(2.5-4)}{(3-1)(3-2)(3-4)} + \\ & + 16 \frac{(2.5-1)(2.5-2)(2.5-3)}{(4-1)(4-2)(4-3)} = 1 * (-0.0625) + 4 * (0.5625) + 9 * (0.5625) + 16 * (-0.0625) = 6.25 \end{aligned}$$

Полученный результат соответствует теории т.е. $f(x) = x^2 = 2.5^2 = 6.25$.

3.3 Интерполяционная формула Лагранжа

Интерполяционный полином Лагранжа может быть записан в другой форме:

$$\Phi_n(x) = \sum_{i=0}^n f(x_i) \prod_{\substack{j=0 \\ j \neq i}}^n \frac{(x-x_j)}{(x_i-x_j)} \quad (23)$$

Запись полинома в виде (23) более удобна для программирования.

При решении задачи интерполяции величина n называется порядком интерполирующего полинома. При этом, как видно из формул (19) и (23), число узлов интерполирования всегда будет равно $n+1$ и значение x , для которого определяется величина $f(x)$, должно лежать внутри области определения узлов интерполяции т.е.

$$x_0 < x_1 < \dots < x_n . \quad (24)$$

В некоторых практических случаях общее известное число узлов интерполяции m может быть больше, чем порядок интерполирующего полинома n .

В этом случае, прежде чем реализовывать процедуру интерполяции согласно формуле (4.5), необходимо определить те узлы интерполяции, для которых справедливо условие (24). При этом следует помнить, что наименьшая погрешность достигается при нахождении значения x в центре области интерполяции. Для обеспечения этого предлагается следующая процедура:

1. После ввода в программу значения величины x необходимо проверить условие $x_0 \leq x \leq x_m$, где x_0 и x_m – начальное и конечное значение узловых точек интерполяции.

2. При выполнении предыдущего условия начинается поиск области интерполяции, для чего находим первое x_i такое, для которого выполняется условие $x_i > x$, при этом номер i будет соответствовать середине интервала интерполяции. Для определения области интерполяции ее левая граница будет начинаться с номера $i - (n+1)/2$, а заканчиваться узлом с номером $i + (n+1)/2$.

3. После выполнения пунктов 1 и 2 программируется формула (23).

Основное назначение интерполяции – это вычисление значений табулированной функции для не узловых (промежуточных) значений аргумента, поэтому интерполяцию часто называют «искусством чтения таблиц между строками».

4 Численное интегрирование

4.1 Общие положения

Очень часто в процессе решения конкретных задач, как в научной, так и в инженерной практике возникает необходимость вычисления определенных интегралов вида:

$$I(f) = \int_a^b f(x) dx.$$

Подынтегральная функция $f(x)$ может быть задана одним из трех способов:

1. Задается явная формула для $f(x)$, например, $f(x) = \cos(xe^{-x^2})$.
2. Функция $f(x)$ явно не задана, но ее значение может быть вычислено при любом x из отрезка a, b . Обычно это значение вычисляется по некоторой подпрограмме.
3. Для некоторого фиксированного конечного набора точек x_i из отрезка a, b задается таблица значений $x_i, f(x_i)$.

Интегралы от функций первого типа иногда удается вычислить аналитически, либо вручную, либо с помощью машинных символьных систем. Интегралы от функций второго и третьего типа (а также первого, если не используются символьные методы) обычно находят численными методами, т.е. методами, позволяющими найти численное значение определенного интеграла приближенно с любой степенью точности.

Все методы приближенного вычисления определенных интегралов основаны на геометрическом смысле интеграла Ньютона-Лейбница. Он заключается в том, что определенный интеграл численно равен площади S криволинейной трапеции, ограниченной графиком функции $f(x)$, осью абсцисс и двумя прямыми $x=a$ и $x=b$. $(aABb)$, как показано на рисунке 3.

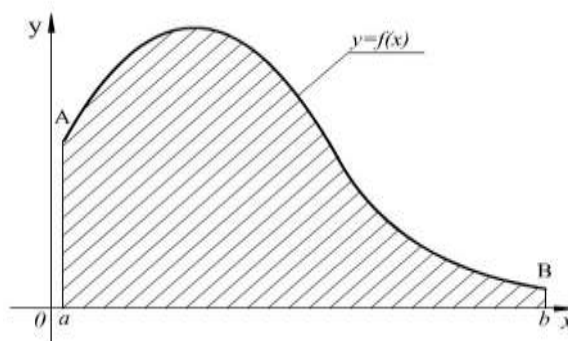


Рисунок 3

Такие методы называют *квадратурными формулами*.

Процедура численного интегрирования заключается в том, что отрезок $[a, b]$ разбивается на n частичных отрезков, а затем подынтегральная функция $f(x)$ аппроксимируется некоторой другой функцией $\phi(x)$, интеграл от которой вычисляется сравнительно просто. Для аппроксимации $f(x)$ может быть использован любой класс простых функций, таких как полиномы, кусочные полиномы, тригонометрические, экспоненциальные или логарифмические функции. Конкретный выбор класса аппроксимирующих функций может зависеть от некоторых определенных свойств подынтегральной функции, но в наиболее распространенном случае, который здесь и рассматривается, в качестве таких функций используются полиномы.

Заменяя подынтегральную функцию на каждом шаге отрезками линий нулевого, первого и второго порядков, получаем соответственно приближенные формулы для вычисления интеграла:

- метод прямоугольников;
- метод трапеций;
- метод Симпсона.

4.2 Метод прямоугольников

Простейшим полиномом является константа. В *формуле прямоугольников* функция $f(x)$ аппроксимируется своим значением в точке a (или в точке b), т.е.

$$I(f) \approx R(f) = (b-a)f(a) \quad (25)$$

Если значение функции $f(x)$ берется в точке a , то формула (25) носит название *формулы левых прямоугольников*.

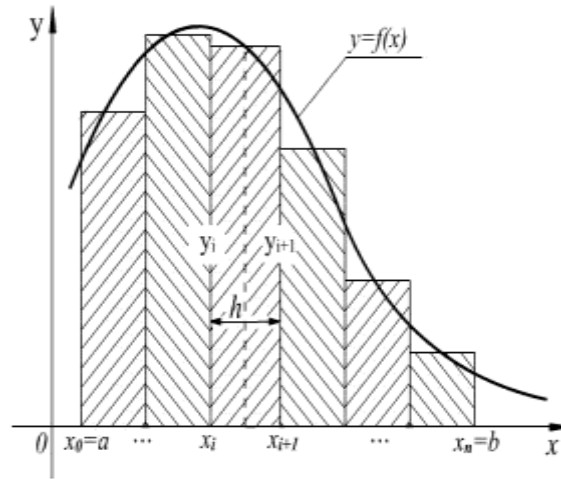


Рисунок 4 – метод средних прямоугольников

Для подсчета интеграла разделим интервал интегрирования a, b на n равных отрезков длины $h = \frac{(b-a)}{n}$. На каждом из отрезков функция $f(x)$ заменяется прямоугольником с отрезками как основаниями, равными h и вертикальными боковыми сторонами высотой $f(x_i)$. При этом точка x_i выбирается, как середина каждого элементарного отрезка. Метод “средних” прямоугольников (метод средних) является более точным, чем методы “левых” и “правых” прямоугольников, когда в качестве точек x_i могут выбираться левые или правые границы элементарных отрезков.

С геометрической точки зрения означает, что площадь криволинейной трапеции $(aABb)$, ограниченной графиком функции $f(x)$, осью абсцисс и двумя прямыми $x=a$ и $x=b$, принимается приближенно равной площади ступенчатой фигуры, образованной из n прямоугольников с основаниями $h = \frac{(b-a)}{n}$ и высотами $f(x_i)$ где $x_i = a + i \cdot h, i = 1, 2, \dots, n$.

Для интервала a, b и шага интегрирования h полная формула будет записана в виде:

$$I(f) = h \sum_{i=0}^{n-1} f(x_i) + \frac{h}{2} \quad (26)$$

где n - число разбиений для интервала $[a, b]$, и точка x_0 совпадает с a .

4.3 Метод трапеции

Следующим простейшим полиномом является линейная функция. Если она выбрана совпадающей с $f(x)$ в концах отрезка a и b , то получаем трапецию.

Площадь этой трапеции (интеграл от линейной функции), используемая в качестве приближения к значению интеграла от $f(x)$, определяется по формуле:

$$I(f) \approx T(f) = \frac{(b-a)}{2} f(a) + f(b) \quad (27)$$

Эта формула (27) известна как *формула трапеции*.

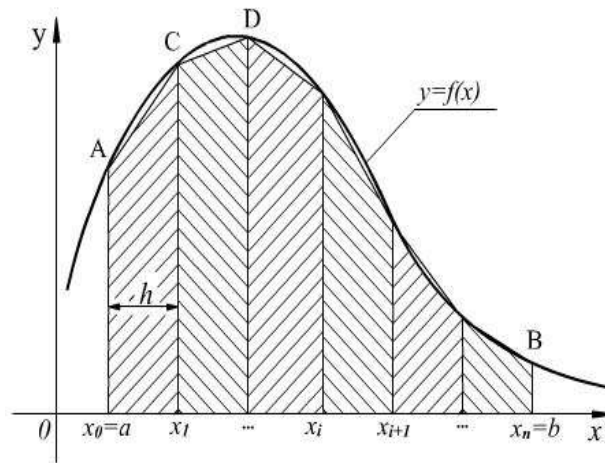


Рисунок 5 – Метод трапеции

Для того чтобы найти приближенное значение площади S , разделим отрезок интегрирования $[a,b]$ на n равных частей длины $h = \frac{(b-a)}{n}$ (рисунок 5). В точках разбиения $x_0 = a, x_1 = a + h, \dots, x_n = b$ проводим ординаты $y_0, y_1, \dots, y_n$ до пересечения с кривой $y = f(x)$, т.е. $y_i = f(x_i)$, $x_i = a + ih, i = 0, 1, 2, \dots, n$. Концы ординат соединяем прямолинейными отрезкам, т.е. на каждом отрезке разбиения дугу графика подынтегральной функции $y = f(x)$ заменяем стягивающей ее хордой (линейная интерполяция), и получим трапецию.

Тогда площадь криволинейной трапеции $aABb$ приближенно можно считать равной площади фигуры, ограниченной ломаной линией $aACD \dots Bb$. Площадь этой фигуры, которую мы обозначим как S , равна сумме площадей трапеций:

$$S = h \cdot \left(\frac{y_0 + y_1}{2} + \frac{y_1 + y_2}{2} + \dots + \frac{y_{n-1} + y_n}{2} \right) =$$

$$= \frac{b-a}{2n} (y_0 + 2y_1 + 2y_2 + \dots + 2y_{n-1} + y_n)$$

Таким образом, для интервала a, b и шага интегрирования h полная формула приближенного значения интеграла будет записана в виде:

$$I(f) = h \left[\frac{(f(a) + f(b))}{2} + \sum_{i=1}^{n-1} f(x_i) \right] \quad (28)$$

где n - число разбиений для интервала a, b и точка x_0 совпадает с a , а точка x_n совпадает с b .

4.4 Метод Симпсона

Более высокая точность определения численного значения определенного интеграла получается при аппроксимации функции $f(x)$ квадратичным интерполяционным полиномом, который совпадает с $f(x)$ в крайних точках a и b , а также в средней точке $\frac{a+b}{2}$. Интеграл от этого квадратичного полинома выражается формулой:

$$I(f) \approx S(f) = \frac{b-a}{6} \left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b) \right] \quad (29)$$

которая называется формулой Симпсона.

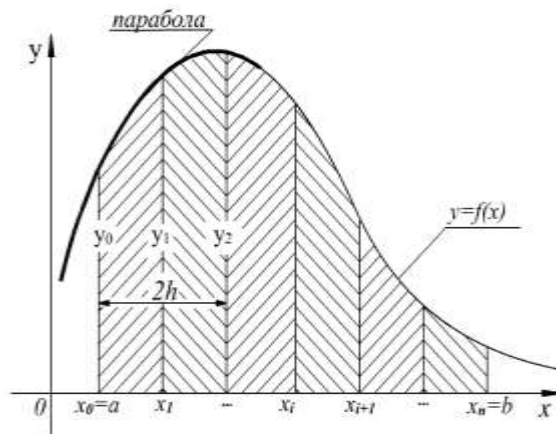


Рисунок 6 – Метод Симпсона

В методе Симпсона площадь криволинейной трапеции рассчитывается как сумма площадей ряда криволинейных трапеций, у которых криволинейная сторона представляет собой участок параболы. Это можно видеть на рисунок 6.

Каждая парабола может быть проведена только через три граничные точки, принадлежащие двум соседним отрезкам. Поэтому число участков разбиения отрезка $[a, b]$ в отличие от предыдущих методов обязательно должно быть четным. Таким образом, вместо каждых двух элементарных прямолинейных трапеций будем рассматривать одну элементарную трапецию, ограниченную параболической дугой. Исходя из этого, определенный интеграл на случай разбиения интервала a, b на n участков с шагом h приближенно вычисляется по формуле:

$$I(f) = \frac{h}{3} \left[f(a) + f(b) + 4 \sum_{i=0}^{(n/2-1)} f(x_{2i+1}) + 2 \sum_{i=1}^{(n/2-1)} f(x_{2i}) \right] \quad (30)$$

(30)– полная формула Симпсона.

Таким образом, для реализации метода прямоугольников, трапеции и Симпсона для вычисления определенного интеграла необходимо:

Задать в явном виде определенный интеграл, площадь которого необходимо определить. После этого задаются пределы интегрирования, и шаг интегрирования. Затем проводится расчет по формулам (26), (28) и (30).

Для метода Симпсона число разбиений n должно быть четным, что подлежит проверке при составлении программы.

ЛИТЕРАТУРА

1. Чечкин А.В. Математическая информатика М: Наука, 1991.
2. Информатика. Базовый курс. Под. Ред. С.В. Симоновича –СпБ: Питер, 2000г.
3. Котлинская Г.П., Галиновский О.И. Программирование на языке Си. - Минск: Высшая школа, 1991.
4. Малышев А.Н. Введение в вычислительную линейную алгебру. - Новосибирск: Наука. Сибирское отделение, 1991.
5. Дэвенпорт Дж. Компьютерная алгебра: Системы и алгоритмы алгебраических вычислений. - М.: Мир, 1991.
6. Амосов А.А., Дубинский Ю.А. Вычислительные методы для инженеров. М: Высшая школа, 1994.
7. Малышев А.Н. Введение в вычислительную линейную алгебру. Новосибирск: Наука Сиб. отделение, 1991.
8. Мудров А.Е. Численные методы для ПЭВМ на языках бейсик, фортран, паскаль - Томск: 1991.
9. Дьяконов В.П. Справочник по MathCad 6.0 – М: СК Пресс, 1997, 328с.
10. Хемминг Р.В. Численные методы. - М.: Наука, 1968.
11. Демидович Б.П. Основы вычислительной математики. - М: Наука, 1966.