

Министерство образования и науки Российской Федерации  
Федеральное агентство по образованию  
Владимирский государственный университет  
Кафедра автоматических и мехатронных систем

# Методические указания по вычислительной практике

Составитель  
Н.А. НОВИКОВА

Владимир 2005

УДК 519.67 (075.8)

ББК 22.192

М 54

Рецензент

Кандидат технических наук, доцент

Владимирского государственного университета

*Н.Р. Рассказчиков*

Печатается по решению редакционно-издательского совета  
Владимирского государственного университета

**М 54**      **Методические указания по вычислительной практике / сост.**  
Н. А. Новикова; Владимирский гос. ун-т. – Владимир : Ред.-издат.  
комплекс ВлГУ, 2005. – 36 с.

Разработаны в соответствии с Государственным образовательным стандартом Министерства образования Российской Федерации.

Предназначены для изучения основ применения вычислительных методов решения различных задач. Значительное внимание уделяется алгоритмам задач решения нелинейных и дифференциальных уравнений, интерполирования и интегрирования функций.

Подготовлены для студентов специальностей 071800 – мехатроника и 210300 – роботы и робототехнические системы очной формы обучения, а также ориентированы на студентов заочной формы обучения и студентов Центра профессионального образования инвалидов.

Табл. 7. Библиогр. : 8 назв.

УДК 519.67 (075.8)

ББК 22.192

## **Введение**

*Методические указания содержат задания, выполняемые студентами во время первой и второй вычислительных практик. Включены основные положения численных методов решения большинства инженерных и технических задач, автоматизации обработки информации. В соответствии с программой практики методические указания содержат следующие разделы: решение нелинейных уравнений, систем линейных уравнений, дифференциальных уравнений, интерполирование и интегрирование функций.*

*В каждом разделе методических указаний приведены: краткие описания теоретических основ решения рассматриваемой задачи, задания и требования к отчету. Как правило, задания требуют решения задачи различными методами с последующим анализом полученных результатов, включающим оценку точности, быстродействия и сходимости. Для выполнения заданий требуется знание типовых приемов программирования на алгоритмическом языке Паскаль.*

*Методические указания предназначены для студентов технических специальностей, изучающих дисциплины, связанные с применением численных методов обработки информации.*

## 1. РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

В практике научных и инженерных расчетов часто возникает необходимость решения уравнений вида

$$f(x) = 0, \quad (1.1)$$

где функция  $f(x)$  определена и непрерывна на некотором конечном или бесконечном интервале  $a \leq x \leq b$ . Если функция представляет многочлен, то уравнение (1.1) называется алгебраическим. Если  $x$  находится под знаком трансцендентной функции (показательной, логарифмической, тригонометрической и т.п.), уравнение (1.1) называется трансцендентным. Значение  $x^*$ , при котором выполняется условие  $f(x^*) = 0$ , называется корнем уравнения (1.1). В общем случае функции  $f(x)$  не имеют аналитических формул для своих корней. В силу этого разработаны численные методы решения уравнений вида (1.1), которые позволяют определять приближенные значения корней с заданной степенью точности.

Процесс отыскания корня уравнения (1.1) состоит из двух этапов:

1) нахождение приближенного значения корня; 2) уточнение приближенного значения до некоторой заданной степени точности.

Первый этап реализуется различными способами. Приближенное значение корня может быть известно, например, из физического смысла задачи. При выделении области, в пределах которой находятся вещественные корни уравнения, можно воспользоваться следующим обстоятельством. Если на концах некоторого отрезка значения непрерывной функции  $f(x)$  имеют разные знаки, то на этом отрезке уравнение  $f(x) = 0$  имеет хотя бы один корень.

Широко распространен графический способ определения приближенных корней. В этом случае строится график функции  $y = f(x)$ , абсциссы точек пересечения которого с осью  $X$  дадут приближенное значение корней. Иногда удается подобрать более простое уравнение, корни которого находятся вблизи корней исходного уравнения. Существует также ряд специальных аналитических методов приближенного нахождения корней многочленов. Найденные приближенные значения корней уточняют различными итерационными методами. Рассмотрим наиболее эффективные из них.

### ***Метод деления пополам***

Для этого метода существенно, чтобы функция  $f(x)$  была непрерывна и ограничена в заданном интервале  $[a, b]$ , внутри которого ищется корень. Предполагается также, что значения функции на концах интервала  $f(a)$  и  $f(b)$  имеют разные знаки, т.е. выполняется условие  $f(a)f(b) < 0$ . Для нахождения корня уравнения  $f(x) = 0$  отрезок  $[a, b]$  делят пополам, т.е. выбирают начальное приближение равным  $x[0] = (a + b)/2$ .

Если  $f(x[0]) = 0$ , то  $x[0]$  является корнем уравнения. В противном случае выбирают тот из отрезков  $[a, x[0]]$  или  $[b, x[0]]$ , на концах которого функция  $f(x)$  имеет разные знаки, ибо корень лежит на этой половине. Интервал вновь делят пополам и выбирают ту половину, на концах которой функция имеет противоположные знаки, и т.д. Если требуется определить корень с точностью  $\varepsilon$ , то деление пополам продолжают до тех пор, пока длина отрезка не даст значение корня с требуемой точностью.

### ***Метод хорд (метод ложного положения)***

В основе этого метода лежит линейная интерполяция по двум значениям функции, имеющим противоположные знаки. При отыскании корня этот метод нередко обеспечивает более быструю сходимость, чем предыдущий. Счет ведется следующим образом. Сначала определяются значения функции в точках, расположенных на оси  $x$  через равные интервалы. Это делается до тех пор, пока не будет найдена пара последовательных значений функции  $f(x_n)$  и  $f(x_{n+1})$ , имеющих противоположные знаки. Прямая, проведенная через эти две точки, пересекает ось  $x$  при значении

$$x^* = x_n - f(x_n) \frac{x_{n+1} - x_n}{f(x_{n+1}) - f(x_n)}.$$

Это значение аргумента используется для определения значения функции  $f(x^*)$ , которое сравнивается со значениями функции  $f(x_n)$  и  $f(x_{n+1})$  и в дальнейшем используется вместо того из них, с которым оно совпадает по знаку. Если значение  $f(x^*)$  недостаточно близко к нулю, то вся процедура повторяется до тех пор, пока не будет достигнута необходимая степень сходимости.

### ***Метод последовательных приближений***

Исходное уравнение можно представить в виде  $x = \varphi(t)$ . Например, можно выделить  $x$  из уравнения, а остальное перенести в правую часть.

Можно также выполнить следующее преобразование:  $x = x + c f(x)$ , где  $c$  – произвольная постоянная. В этом случае  $\varphi(x) = x + c f(x)$ . Задаются начальным приближением  $x[0]$ , а последующие приближения определяют итерационной процедурой вида  $x[k+1] = \varphi(x[k])$ ,  $k = 0, 1, 2, \dots$ . Эта итерационная процедура сходится, если на отрезке  $[a, b]$ , содержащем корень  $x^*$ , а также все его последовательные приближения  $x[0], x[1], \dots, x[k]$ , выполнено условие  $|\varphi'(x)| < 1$ , Сходимость будет тем быстрее, чем меньше по абсолютной величине значение производной  $\varphi'(x)$ .

Метод последовательных приближений обладает тем важным преимуществом, что при его использовании не накапливаются ошибки вычислений. Ошибка вычислений эквивалентна ухудшению очередного приближения, а это может отразиться только на числе итераций, но не на точности результата.

### ***Метод Ньютона***

Пусть уравнение  $f(x) = 0$  имеет один корень на отрезке  $[a, b]$ , причем первая и вторая производные  $f'(x)$ ,  $f''(x)$  определены, непрерывны и сохраняют постоянные знаки на отрезке  $[a, b]$ .

Выбирают некоторое начальное приближение корня  $x[0]$  на интервале  $[a, b]$  и проводят касательную в точке  $C_0(x[0], f(x[0]))$  к кривой  $y = f(x)$  до пересечения с осью абсцисс в точке  $x[1]$ . Абсциссу  $x[1]$  принимают за очередное приближение корня. Уравнение касательной в точке  $C_0$  имеет вид

$$y = f(x[0]) + f'(x[0])(x - x[0]). \quad (1.2)$$

Полагая в (1.2)  $y = 0$ , находят абсциссу  $x[1]$ :

$$x[1] = x[0] - \frac{f(x[0])}{f'(x[0])}.$$

Далее проводят касательную через новую точку  $C_1(x[1], f(x[1]))$  и находят точку ее пересечения с осью абсцисс  $x[2]$ :

$$x[2] = x[1] - \frac{f(x[1])}{f'(x[1])}.$$

Эту точку принимают за новое приближение корня. Аналогично находят последующие приближения:

$$x[k] = x[k-1] - \frac{f(x[k-1])}{f'(x[k-1])}.$$

Итерационный процесс прекращают при выполнении условий:

$$\delta \leq \varepsilon, f(x[k]) \leq 100\varepsilon;$$

$$\delta = \begin{cases} \left| \frac{x[k] - x[k-1]}{x[k]} \right|, & \text{если } |x[k]| > 1; \\ |x[k] - x[k-1]|, & \text{если } |x[k]| \leq 1; \end{cases}$$

где  $\varepsilon$  – заданная погрешность вычислений.

Начальное приближение  $x[0]$  целесообразно выбирать из условия  $f(x[0])f''(x[0]) > 0$ . В противном случае сходимость метода Ньютона не гарантируется. Чаще всего выбирают  $x[0] = a$  или  $x[0] = b$  в зависимости от того, для какой из точек выполняется указанное условие.

Метод Ньютона эффективен для решения уравнений, у которых значение модуля производной  $|f'(x)|$  близ корня достаточно велико, т.е. график функции  $y = f(x)$  в окрестности корня имеет бóльшую крутизну. В данном методе погрешность очередного приближения примерно равна квадрату погрешности предыдущего приближения. Сходимость этого метода существенно выше сходимости метода последовательных приближений. Метод Ньютона чаще других применяют для нахождения корней произвольной дифференцируемой функции.

Отметим, что все сказанное справедливо, если начальное приближение  $x[0]$  выбрано достаточно близко к истинному корню уравнения, что не всегда осуществимо. В силу этого методу Ньютона часто предшествует какой-либо надежно сходящийся алгоритм (например метод деления пополам). Метод Ньютона в таком случае работает на завершающей стадии решения уравнения.

### ***Метод секущих***

Один из недостатков метода Ньютона состоит в том, что, пользуясь им, приходится дифференцировать функцию  $f(x)$ . Если нахождение производной затруднено, то можно воспользоваться некоторым приближени-

ем, которое и составляет основу метода секущих. Заменяв производную  $f'(x)$ , используемую в методе Ньютона (1.2) разностью последовательных значений функции, отнесенной к разности значения аргумента

$$F'(x[k]) = \frac{f(x[k]) - f(x[k-1])}{x[k] - x[k-1]},$$

получим следующую итерационную формулу:

$$x[k+1] = x[k] - \frac{f(x[k])}{F'(x[k])}. \quad (1.3)$$

Схема алгоритма для этого метода та же, что и для метода Ньютона (несколько иной вид имеет итерационная формула).

В сущности, в методе секущих для отыскания корня используется комбинация интерполяции и экстраполяции. В своей интерполяционной части этот метод эквивалентен методу ложного положения. Как и в случае метода Ньютона, счет заканчивается, когда последовательные значения совпадают с некоторой приемлемой точностью или когда значение функции  $f(x)$  становится достаточно близким к нулю. В случае кратных корней при использовании метода секущих возникают те же трудности, что и при использовании метода Ньютона.

### **Задание**

1. Составить схемы алгоритмов решения нелинейных уравнений методами деления пополам, последовательных приближений и Ньютона. При составлении программ предусмотреть определения числа итераций для получения решения с заданной точностью.

2. Разработать программы решения нелинейных уравнений с использованием подпрограмм.

3. Решить нелинейные уравнения, приведенные в табл. 1.1. Начальное приближение корня выбрать из указанной области. Параметр  $\varepsilon$  принять равным 0.001.

4. На основании результатов расчетов п. 3 провести сравнительный анализ методов.









щих большие значения. В таких случаях следует пользоваться нормированными корректирующими приращениями

$$\frac{\Delta x_i}{x_i} \leq 0,0001, \text{ где } i = 1, 2, \dots, n.$$

Для системы двух уравнений

$$\begin{cases} f(x, y) = 0 \\ g(x, y) = 0 \end{cases}$$

согласно методу Ньютона последовательные приближения приращений  $\Delta x_{n+1}, \Delta y_{n+1}$  вычисляются по формулам

$$\Delta x_{n+1} = -\frac{1}{J(x_n, y_n)} \begin{vmatrix} f(x_n, y_n) & f'_y(x_n, y_n) \\ g(x_n, y_n) & g'_y(x_n, y_n) \end{vmatrix},$$

$$\Delta y_{n+1} = -\frac{1}{J(x_n, y_n)} \begin{vmatrix} f'_x(x_n, y_n) & f(x_n, y_n) \\ g'_x(x_n, y_n) & g(x_n, y_n) \end{vmatrix},$$

где  $f'_y(x_n, y_n)$  и  $f'_x(x_n, y_n)$ ,  $g'_y(x_n, y_n)$ ,  $g'_x(x_n, y_n)$  частные производные функций  $f(x, y)$  и  $g(x, y)$  по переменным  $y$  и  $x$  соответственно в точке начального приближения  $(x_n, y_n)$ ,  $J(x_n, y_n)$  – якобиан в точке  $(x_n, y_n)$

$$J(x_n, y_n) = -\frac{1}{J(x_n, y_n)} \begin{vmatrix} f'_x(x_n, y_n) & f'_y(x_n, y_n) \\ g'_x(x_n, y_n) & g'_y(x_n, y_n) \end{vmatrix}.$$

Хотя метод Ньютона имеет преимущества по сравнению с другими итерационными методами, для него также существует проблема сходимости. Величина области сходимости обратно пропорциональна степени сложности и числу уравнений.

### Задание

1. Составить схемы алгоритмов решения нелинейных систем уравнений методами Ньютона и простой итерации.
2. Разработать программы решения систем нелинейных уравнений методами Ньютона и простой итерации с использованием подпрограмм.
3. Решить системы нелинейных уравнений, приведенных в табл. 2.1. Параметр  $\varepsilon$  принять равным 0,00001.
4. Провести сравнительный анализ методов.

## Содержание отчета

1. Схемы алгоритмов и программ.
2. Листинг программы.
3. Протоколы результатов решения задач.
4. Выводы по работе.

Таблица 2.1

№	Система уравнений	$x_1^0, x_2^0$	№	Система уравнений	$x_1^0, x_2^0$
1	$\begin{cases} \cos \frac{x_1 x_2}{6} - x_2 + 0.5 = 0 \\ \ln\left(1 + \frac{x_1 + x_2}{5}\right) + 1.1 = \sin \frac{x_2}{3} + x_1 \end{cases}$	1,1	4	$\begin{cases} x_1^3 + x_2^3 - 6x_1 + 3 = 0 \\ x_1^3 - x_2^3 - 6x_2 + 2 = 0 \end{cases}$	0,5 0,5
2	$\begin{cases} \lg \frac{x_2}{x_3} - x_1 + 1 = 0 \\ 2x_1^2 + x_2 - x_3^2 - 0.4 = 0 \\ \frac{x_1 x_2}{20} - x_3 + 2 = 0 \end{cases}$	1, 2,2 2	5	$\begin{cases} x_1 + 3 \lg x_1 - x_2^2 = 0 \\ 2x_1 - x_1 x_2 - 5x_1 + 1 = 0 \end{cases}$	3,4 2,2
3	$\begin{cases} x_1 + x_1^2 - 2x_2 x_3 - 0.1 = 0 \\ x_2 - x_2^2 + 3x_1 x_3 + 0.2 = 0 \\ x_3 + x_3^2 + 2x_1 x_2 - 0.3 = 0 \end{cases}$	0 0 0	6	$\begin{cases} 2x_1 + x_2 - 10 \lg x_2 = 4 \\ 5x_1 - 6x_2 + 20 \lg x_1 = -16 \end{cases}$	1,1
			7	$\begin{cases} 0.3 \cos x_1 - x_2 = 0 \\ 0.5 \sin \frac{x_2}{3} - x_1 + 1 = 0 \end{cases}$	1,0

### 3. РЕШЕНИЕ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Уравнения, содержащие суммы целых степеней  $x$  называют алгебраическими. Их общий вид

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \dots + a_1 x + a_0 = 0.$$

При отыскании корней алгебраических уравнений полезно иметь в виду некоторые их свойства:

1. Алгебраическое уравнение порядка  $n$  имеет  $n$  корней, которые могут быть действительными или комплексными.

2. Если все коэффициенты  $a_i$  действительные, то все комплексные корни образуют комплексно-сопряженные пары.

3. Число положительных действительных корней равно или меньше (на целое число) числа перемен знаков в последовательности коэффициентов  $a_i$ .

4. Число отрицательных действительных корней равно или меньше (на целое число) числа перемен знаков в последовательности коэффициентов  $a_i$ , при замене  $x$  на  $-x$ .

Известны прямые методы отыскания корней алгебраических уравнений второй и третьей степеней, однако для уравнений более высоких степеней приходится использовать итерационные методы. Для нахождения действительных или комплексных корней алгебраических уравнений можно использовать алгоритмы решения трансцендентных уравнений, если использовать комплексные числа.

### *Метод Лина*

Существует несколько специальных методов для отыскания комплексных корней. Все они почти всегда связаны с выделением из исходного алгебраического уравнения квадратичного множителя  $x^2 + px + q$ .

Широко применяемым методом этого типа является метод Лина. В его основе лежит представление алгебраического уравнения

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0 \quad (3.1)$$

в виде

$$(x^2 + px + q)(b_n x^{n-2} + b_{n-1} x^{n-3} + \dots + b_3 x + b_2) + b_1 x + b_0 = 0, \quad (3.2)$$

где  $b_1 x + b_0$  - линейный остаточный член, который мы стремимся сделать равным нулю.

Если это возможно, то исходное алгебраическое уравнение делится на квадратичный множитель без остатка. Предположив, что  $b_1 = b_0 = 0$  и сравнивая соответствующие коэффициенты двух приведенных форм алгебраического уравнения, получаем:

$$\begin{aligned} b_n &= a_n; \\ b_{n-1} &= a_{n-1} - p b_n; \\ b_{n-2} &= a_{n-2} - p b_{n-1} - q b_n; \\ &\dots\dots\dots \\ b_{n-j} &= a_{n-j} - p b_{n+1-j} - q b_{n+2-j}; \\ p &= (a_1 - q b_3) / b_2; \quad q = a_0 / b_2. \end{aligned} \quad (3.3)$$

Алгоритм итерационной процедуры решения действует следующим образом. Сначала задаются вероятные значения  $p$  и  $q$ , которые вместе с

заданными коэффициентами  $a_i$  используются для вычисления  $b_{n-1}$ . В свою очередь, значение  $b_{n-1}$  служит для вычисления  $b_{n-2}$  и т.д. до тех пор, пока не будут найдены все коэффициенты, включая  $b_2$ , по формулам (3.3). Значения  $b_3, b_2, a_1$  и  $a_0$  позволяют найти из двух последних уравнений (3.3) уточненные значения  $p$  и  $q$ , которые обозначим через  $p^*$  и  $q^*$ . Если их отличие от  $p$  и  $q$  достаточно мало, счет прекращается. В противном случае вся процедура повторяется с использованием новых значений  $p$  и  $q$ . Процедура вычислений сводится к решению системы двух уравнений с двумя неизвестными методом простой итерации.

### Задание

1. Составить схему алгоритма решения алгебраического уравнения порядка  $n$   $a_n x^n + a_{n-1} x^{n-1} + \dots + a_0 = 0$  с произвольными коэффициентами  $a_i (i = 0, \dots, n)$  методом Лина.

2. Разработать программу решения алгебраического уравнения общего вида методом Лина с использованием подпрограмм.

3. Решить контрольный пример из табл. 3.1.

Таблица 3.1

№ п/п	$n$	Коэффициенты $a_n, a_{n-1}, \dots, a_0$
1	4	2, 0.2, 1.4, 2.3, -1.5
2	4	1, 1, 0, 2, 1
3	4	1, 2, 1, 2, 1
4	4	1, 4, 4, 4, -1
5	4	1, 4, 2, 2, 6.4, 2.56
6	4	1, -8, -5, -12, 2.25
7	4	1, 10, 1.8, -31, 9.61

### Содержание отчета

1. Схемы алгоритмов и программ.
2. Листинги программ.
3. Протокол результатов решения задачи.
4. Выводы.





Прямые методы используются при решении систем небольшого порядка ( $n > 10^3$ ). Итерационные методы выгодно применять для систем высокого порядка ( $n \approx 10^3 - 10^6$ ) со слабо заполненной матрицей коэффициентов.

***Метод Якоби (метод простой итерации)***

Итерационные схемы решения систем уравнений применяются к системам, предварительно приведенным к виду

$$\left. \begin{aligned}
 x_1 &= C_{12}x_2 + C_{13}x_3 + \dots + C_{1n}x_n + d_1; \\
 x_2 &= C_{21}x_1 + C_{23}x_3 + \dots + C_{2n}x_n + d_2; \\
 &\dots\dots\dots \\
 x_n &= C_{n1}x_1 + C_{n2}x_2 + \dots + C_{nn-1}x_{n-1} + d_n.
 \end{aligned} \right\} \tag{4.1}$$

Применяемые для решения таких уравнений методы – это методы Якоби и Зейделя, в основе которых лежит систематическое уточнение значений переменных, заданных в начале счета.

В методе Якоби исходные значения переменных используются для вычислений новых значений  $x_1, x_2, \dots, x_n$  с помощью приведенных выше уравнений (4.1). Процесс прекращается, когда все новые значения оказываются достаточно близкими к исходным. В противном случае новые значения используются вместо исходных. Этот процесс повторяют до тех пор, пока не будет достигнута сходимость или не станет ясно, что процесс расходится. В этом методе замена значений всех переменных производится одновременно (одновременное смещение).

***Метод Зейделя***

Итерационный метод Зейделя обеспечивает возможность, задавшись некоторым произвольным вектором  $x[0]$  (начальное приближение к искомому решению  $x^*$ ), построить последовательность приближений значений  $x[k], k = 1, 2, \dots$ , сходящихся к точному решению системы  $x^*$ . Алгоритм метода состоит в следующем.

1. Исходную систему уравнений решают относительно неизвестных  $x_1, x_2, \dots, x_n$ , т.е. приводят к виду (4.1). Такое преобразование всегда выполнимо для системы, определитель которой не равен нулю, и может быть осуществлено различными способами.

2. Задают начальное приближение  $x[0] = \{x_1[0], \dots, x_n[0]\}$ . Начальный вектор может быть выбран произвольно, однако необходимо использовать всю имеющуюся информацию о системе уравнений, чтобы  $x[0]$  располагался как можно ближе к точному решению системы  $x^*$ .

3. В первое уравнение системы подставляют координаты точки  $x[0]$  и вычисляют новое значение первой координаты:

$$x_1[1] = C_{11}x_1[0] + C_{12}x_2[0] + \dots + C_{1n}x_n[0] + d_1.$$

Применяя вычисленное значение  $x_1[1]$  и начальное значение остальных переменных  $x_2[0], \dots, x_n[0]$ , из второго уравнения системы определяют новое значение второй координаты:

$$x_2[1] = C_{21}x_1[1] + C_{22}x_2[0] + \dots + C_{2n}x_n[0] + d_2.$$

Используя аналогично уже вычисленные приближения, получают значения всех остальных координат. Так, последняя координата будет иметь значение:

$$x_n[1] = C_{n1}x_1[1] + C_{n2}x_2[1] + \dots + C_{nn-1}x_{n-1}[1] + C_{nn}x_n[0] + d_n.$$

В результате будет определено первое приближение  $x[1]$  к решению системы  $x^*$ .

4. Начальный вектор  $x[0]$  заменяют вектором  $x[1]$  и вычисляют следующее приближение. В общем случае  $k+1$  приближение определяют по формулам:

$$x_1[k+1] = C_{11}x_1[k] + C_{12}x_2[k] + \dots + C_{1n}x_n[k] + d_1;$$

$$x_2[k+1] = C_{21}x_1[k+1] + C_{22}x_2[k] + \dots + C_{2n}x_n[k] + d_2;$$

.....

$$x_n[k+1] = C_{n1}x_1[k+1] + C_{n2}x_2[k+1] + \dots + C_{nn}x_n[k] + d_n.$$

Итерационный процесс продолжают до тех пор, пока все  $x_i[k+1]$  не станут достаточно близкими к  $x_i[k]$ . Итерации прекращают при выполнении условия

$$\max \frac{x_i[k+1] - x_i[k]}{x_i[k+1]} < \varepsilon, \quad i = 1, 2, \dots, n,$$

где  $\varepsilon$  – некоторое заданное положительное число (точность вычислений).

Весьма важно, что итерационные методы являются самоисправляющимися, т.е. отдельная ошибка, допущенная при вычислениях, не отражается на конечном результате, так как ошибочное приближение рассматривается как новый начальный вектор.

Важным преимуществом итерационных методов является удобство их программирования, так как они требуют выполнения однообразных повторяющихся операций.

### Задание

1. Составить схемы алгоритмов решения систем линейных уравнений методами Якоби и Зейделя. Предусмотреть контроль количества итераций.

2. Разработать программу решения систем линейных уравнений методами Якоби и Зейделя с использованием подпрограмм.

3. Решить системы линейных уравнений, приведенные в табл. 4.1, с точностью до 0,001.

4. Вычислить точные оценки погрешности методов по координатам  $\delta = \max |x_i - x_i^*|$ ,  $i = 1, 2, \dots, n$  ( $x$  в табл. 4.1 – точное решение системы).

5. На основании полученных результатов провести сравнительный анализ методов по точности и времени решения.

Таблица 4.1

№	Матрица А	Вектор В	x	№	Матрица А	Вектор В	x
1	10.9 1.2 2.1 0.9	-7	-1	4	6.1 2.2 1.2	16.55	1.5
	1.2 11.2 1.5 2.5	5.3	0		2.2 5.5 -1.5	10.55	2
	2.1 1.5 9.8 1.3	10.3	1		1.2 -1.5 7.2	16.8	2.5
	0.9 2.5 1.3 12.1	24.6	2				
2	20.9 1.2 2.1 0.9	21.77	0.8	5	0.15 2.11 30.75	-26.38	1
	1.2 21.2 1.5 2.5	27.46	1		0.64 1.21 2.05	1.01	2
	2.1 1.5 19.8 1.3	28.76	1.2		3.21 1.53 1.04	5.23	-2
	0.9 2.5 1.3 32.1	49.72	1.4	6	1.02 -0.25 -0.3	0.515	2
		2.5	-0.41 1.13 -0.15		1.555	2.5	
3	3.82 1.02 0.75 0.81	15.655	3	7	-0.25 -0.14 1.21	2.78	3
	1.05 4.53 0.98 1.53	22.705	3.5		1.15 0.42 100.71	198.7	2
	0.73 0.85 4.71 0.81	23.48	2	1.19 0.55 0.32	2.29	1	
	0.88 0.81 1.28 3.5	16.11		1.00 0.35 3.00	-0.65	-2	

### Содержание отчета

1. Схемы алгоритмов и программ.
2. Листинги программы.
3. Протокол результатов решения.
4. Результаты расчетов.
5. Выводы.

## 5. ИНТЕРПОЛИРОВАНИЕ ФУНКЦИЙ

Данные, с которыми приходится иметь дело инженеру, часто представляются в виде таблиц. Это может быть связано либо с тем, что результаты были получены экспериментально и лишь для некоторых дискретных значений аргумента, либо с тем, что объем таблиц ограничен и в них можно привести лишь некоторые данные. Сущность интерполяции состоит в отыскании значения функции в некоторой промежуточной точке.

### *Линейная интерполяция*

Простейшим видом интерполяции является линейная интерполяция, в основе которой лежит аппроксимация кривой на участке между точками  $(x_k, y_k)$  и  $(x_{k+1}, y_{k+1})$  прямой, проходящей через те же точки. Уравнение прямой можно представить в виде

$$\frac{y - y_k}{x - x_k} = \frac{y_{k+1} - y_k}{x_{k+1} - x_k}$$

или в виде

$$y = \frac{y_k(x - x_{k+1}) - y_{k+1}(x - x_k)}{x_k - x_{k+1}}.$$

Таким образом, зная два табличных значения  $y_k$  и  $y_{k+1}$  соответствующих  $x_k$  и  $x_{k+1}$ , с помощью указанных формул можно найти значение функции  $y$  при любом значении  $x$  в интервале  $[x_k, x_{k+1}]$ . Обычно полагают, что, используя большее число соседних точек и аппроксимируя истинную кривую более сложной линией, можно уточнить полученный результат. Рассмотрим методы нахождения единственного многочлена  $n$ -й степени  $P_n(x)$ , аппроксимирующего функцию  $f(x)$  кривой, проходящей через все  $n+1$  заданные в таблице точки  $(x_i, y_i)$ , где  $i = 0, 1, \dots, n$ . В этом случае говорят, что многочлен удовлетворяет условиям

$$P_n(x_i) = y_i, \quad i = 0, 1, \dots, n.$$

Методы отыскания такого многочлена делятся на три группы: методы Лагранжа, разностные методы и итерационные.

### *Интерполяция по Лагранжу*

При этой интерполяции задается  $n+1$  табличное значение  $(x_i, y_i)$ , где  $i = 0, 1, \dots, n$ . Предполагается, что точки  $x_i, y_i$  принадлежат кривой  $y = f(x)$



вестный также как метод Ньютона-Грегори. Интерполяционный многочлен для этого метода имеет вид:

$$P_n(x) = C_0 + C_1(x - x_0) + C_2(x - x_0)(x - x_1) + \dots + C_n(x - x_0)(x - x_1)\dots(x - x_{n-1}).$$

Коэффициенты  $C_j$  находятся из уравнений  $P_n(x_i) = Y_i$  или

$$C_0 = Y_0,$$

$$C_0 + C_1(x - x_0) = Y_1,$$

$$C_0 + C_1(x - x_0) + C_2(x_2 - x_0)(x_2 - x_1) = Y_2,$$

.....

$$C_0 + \dots + C_n(x_n - x_0)(x_n - x_1)\dots(x_n - x_{n-2}) = Y_n.$$

Это линейная система уравнений с треугольной матрицей и определение с ее помощью значений  $C_j$  не вызывает затруднений, однако существует и еще более простой способ определения  $C_j$ , основанный на применении правых конечных разностей.

Если значения  $x$  заданы через равные промежутки  $x_{i+1} - x_i = h$ , то в общем случае  $x_i = x_0 + ih$ , где  $i = 1, 2, \dots, n$ . Последнее выражение позволяет привести решаемые уравнения к виду

$$Y_0 = C_0,$$

$$Y_1 = C_0 + C_1h,$$

$$Y_2 = C_0 + C_1(2h) + 2h^2C_2,$$

.....

$$Y_i = C_0 + C_1(ih) + C_2ih[(i-1)h] + \dots + C_i(i!)h^i,$$

откуда для коэффициентов получаем

$$C_0 = Y_0$$

$$C_1 = \frac{Y_1 - C_0}{h} = \frac{Y_1 - Y_0}{h} = \frac{\Delta Y}{h}.$$

Здесь  $\Delta Y$  называется первой правой разностью. Продолжая, находим

$$C_2 = \frac{1}{2h^2}(Y_2 - C_0 - 2hC_1) = \frac{1}{2h^2}[(Y_2 - Y_1) - (Y_1 - Y_0)] = \frac{1}{2h^2}[\Delta(\Delta Y_0)] = \frac{\Delta^2 Y_0}{2h^2},$$

где  $\Delta^2 Y_0$  – вторая правая разность, представляющая собой разность первых разностей. Коэффициент  $C_j$  можно представить в виде:

$$C_j = \frac{\Delta^j Y_0}{(j!)h^j}.$$

В общем случае разности более высоких порядков для функции  $y = f(x)$  в интервале  $x_0 \leq x \leq x_n$  определяются выражением

$$\Delta^j Y_j = \Delta^{j-1} Y_{j+1} + \Delta^{j-1} Y_j, \quad i = 0, 1, \dots, n - j.$$

### ***Итерационные методы интерполяции***

Эти методы основаны на повторном применении простой интерполяционной схемы. Наиболее известным из них является метод Эйткена, сущность которого в повторном применении интерполяции.

Выше было показано, что линейная интерполяция между точками  $(x_0, y_0)$  и  $(x_i, y_i)$  осуществляется по формуле:

$$Y_{i1}(x) = \frac{1}{x_i - x_0} [Y_0(x_i - x) - Y_i(x_0 - x)],$$

с помощью которой, задав значение  $x_i$ , можно составить таблицу функций  $y_{i1}(x)$ , где  $i = 1, 2, \dots, n$ .

Пользуясь этими функциями, с помощью линейной интерполяции

$$Y_{i2}(x) = \frac{1}{x_i - x_1} [Y_{11}(x)(x_i - x) - Y_{i1}(x)(x_1 - x)],$$

получим новое семейство соотношений. Простой подстановкой можно показать, что выражения для  $Y_{i2}(x)$  представляют собой многочлены второй степени, описывающие кривые, проходящие через точки  $(x_0, y_0)$ ,  $(x_1, y_1)$  и  $(x_i, y_i)$ . Получив многочлены  $Y_{i2}$  с помощью линейной интерполяции и используя функции  $Y_{i2}(x)$ , можно записывать выражение для многочлена третьей степени

$$Y_{i3}(x) = \frac{1}{x_i - x_2} [Y_{22}(x)(x_i - x) - Y_{i2}(x)(x_2 - x)],$$

описывающего кривые, проходящие через точки  $(x_0, y_0)$ ,  $(x_1, y_1)$ ,  $(x_2, y_2)$  и  $(x_i, y_i)$ . Продолжая этот процесс, получим значения  $Y_{ij}(x)$ , которые будут стремиться к значению  $f(x)$ . Хотя этот метод позволяет вводить многочлены степени  $n > 3$ , обычно этого не делают, стремясь избежать роста ошибок. Следует, однако, отметить, что метод Эйткена не требует, чтобы используемые для интерполяции значения функции были расположены через равные интервалы.

## Задание

1. Составить схемы алгоритмов интерполирования функций по методам, указанным преподавателем.
2. Разработать программу интерполирования функций.
3. Выполнить интерполирование функций, приведенных в табл. 5.1.
4. Построить графики функций.

Таблица 5.1

№ п/п	Значение аргумента	Значение функций	Пределы изменения аргумента	Шаг	№ п/п	Значение аргумента	Значение функций	Пределы изменения аргумента	Шаг
1	-1,06 -0,837 -0,684 -0,315 -0,117 0,0 0,115 0,5	1,22 0,854 0,513 0,271 0,217 0,198 0,218 0,277	-1,8 0,4	0,2	4	0,215 0,441 0,638 0,865 1,05 1,3 1,55 1,82	5,82 4,63 4,1 3,34 3,0 3,29 4,32 5,72	0,4 1,6	0,2
2	-2,15 -1,83 -1,62 -1,45 -1,01 -0,72 -0,48 0,0	-2,23 -2,65 -3,1 -3,54 -4,26 -4,38 -4,52 -4,27	2,95 0,1	0,3	5	-1,0 -0,96 -0,86 -0,79 0,22 0,5 0,93 1,1	-1,0 -0,151 0,894 0,986 0,895 0,5 -0,306 -0,51	-1,01,02	0,35
3	-0,21 -0,143 -0,099 -0,032 0,114 0,182 0,257 0,38	-12,64 -11,05 -10,25 -9,32 -9,25 -10,0 -11,48 -14,4	-0,21 0,28	0,08	6	0,5 0,75 1,0 1,25 1,5 1,75 2,0 2,25	0,1915 0,2734 0,3413 0,3944 0,4332 0,4599 0,4773 0,4878	0,625 2,226	0,25

## Содержание отчета

1. Схемы алгоритмов и программ.
2. Листинг программы.
3. Протокол результатов решения задачи.
4. Графики функций.
5. Выводы.



## 6. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ ФУНКЦИЙ

Во многих случаях задачи вычисления определенных интегралов не могут быть решены точно, так как не удается выразить первообразную функцию через элементарные функции. Довольно часто нахождение первообразной связано с необходимостью выполнения весьма сложных преобразований. Распространенной является также ситуация, когда подынтегральная функция задана таблицей экспериментально полученных значений. Во всех этих случаях прибегают к приближенному вычислению определенных интегралов.

В основе численного интегрирования лежит приближенное вычисление значения определенного интеграла на основании ряда значений подынтегральной функции. В общем случае задача формулируется как нахождение значения

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

Обычный прием состоит в том, что функцию  $f(x)$  на рассматриваемом отрезке  $[a, b]$  заменяют интерполирующей или аппроксимирующей функцией  $\varphi(x)$  простого вида (например, полиномом), а затем приближенно полагают

$$\int_a^b f(x) dx = \int_a^b \varphi(x) dx$$

Функция  $\varphi(x)$  должна быть такой, чтобы интеграл  $\int_a^b \varphi(x) dx$  вычислялся непосредственно.

### *Метод трапеций*

Пусть  $y = f(x)$  – непрерывная функция в интервале  $[a, b]$ . Требуется

вычислить определенный интеграл  $I = \int_a^b f(x) dx$ , где  $a, b$  конечны. Интер-

вал интегрирования  $[a, b]$  разбивается на  $n$  равных частей, имеющих длину  $h = (b - a)/n$ . Площадь под кривой  $y = f(x)$  на одном из участков разбиения  $[x_i, x_{i+1}]$  определяется выражением

$$I_i = \int_{x_i}^{x_{i+1}} f(x) dx.$$

Эту площадь можно с некоторой погрешностью считать равной площади трапеции

$$I_i \approx \frac{h(f(x_i) + f(x_{i+1}))}{2}.$$

Общая площадь равна сумме площадей на отдельных участках разбиения, т. е.

$$I = \int_a^b f(x) dx = \sum_{i=1}^n \int_{x_i}^{x_{i+1}} f(x) dx \approx \frac{h}{2} \sum_{i=0}^{n-1} (f(x_i) + f(x_{i+1})) =$$

$$= \frac{h}{2} (f(x_0) + 2f(x_1) + \dots + 2f(x_{n-1}) + f(x_n)) = \left( \frac{f(a) + f(b)}{2} + f(x_1) + \dots + f(x_{n-1}) \right) h,$$

где  $x_1 = a + h$ ;  $x_2 = a + 2h, \dots, x_{n-1} = a + (n-1)h$ .

Погрешность вычислений определенного интеграла по методу трапеций оценивают согласно формуле:

$$|\Delta| \leq \frac{(b-a)^3}{12n^2} M, \text{ где } M = \max_{a \leq x \leq b} |f''(x)|.$$

### **Метод Симпсона**

Интервал интегрирования  $[a, b]$  разбивается на четное число  $2n$  равных частей длиной  $h = (b - a)/2n$ . Через три последовательные ординаты разбиения на концах двух соседних отрезков проводится квадратичная парабола и вычисляется площадь получения фигуры. Общая площадь обра-

зуется путем суммирования площадей отдельных фигур и определяется по формуле:

$$I = \int_a^b f(x)dx \approx \frac{h}{3}(y_0 + 4(y_1 + y_3 + \dots + y_{2n-1}) + 4(y_2 + y_4 + \dots + y_{2n-2}) + y_{2n}),$$

где  $x_k = a + kh$ ;  $y_k = f(x_k)$ ,  $k = 0, 1, \dots, 2n$ .

Погрешность данного метода можно оценить по формуле:

$$|\Delta| \leq \frac{(b-a)^5}{180 \cdot 12n^4} M,$$

где  $M = \max_{a \leq x \leq b} |f^{iv}(x)|$ .

Метод Симпсона обладает достаточно высокой для практических вычислений точностью, прост, хорошо программируется, вследствие чего он широко применяется.

### *Метод Гаусса*

В данном методе концы участков разбиения интервала  $[a, b]$  располагают таким образом, чтобы при заданном количестве участков  $n$  добиться наивысшей точности интегрирования. Можно показать, что этот метод точно интегрирует все многочлены степени, меньшей или равной  $2n - 1$ .

Предварительно интервал интегрирования  $[a, b]$  преобразуется в интервал  $[-1, 1]$  путем замены переменных:

$$x = \frac{b-a}{2}t + \frac{b+a}{2}; \quad \varphi(t) = \frac{b-a}{2} f\left(\frac{b-a}{2}t + \frac{b+a}{2}\right).$$

В результате получают  $I = \int_{-1}^1 \varphi(t)dt$ .

В методе Гаусса используется квадратурная формула вида:

$$\int_{-1}^1 \varphi(t)dt \approx \sum_{k=0}^n a_k \varphi(t_k).$$

Абсциссы  $t_k$  называются узлами, а коэффициенты  $a_k$  – весовыми коэффициентами квадратурной формулы. Значения  $t_k$  являются корнями по-

линомов Лежандра степени  $n$ . Полиномы Лежандра  $P_0(t)$  можно отыскать с помощью выражений:

$$\left. \begin{aligned} P_0(t) &= 1; \\ P_1(t) &= t; \\ P_m(t) &= \frac{1}{m}((2m-1)tP_{m-1}(t) - (m-1)P_{m-2}(t)). \end{aligned} \right\}$$

Весовые коэффициенты можно найти из соотношения

$$a_k = \frac{2}{(1-t_k^2)(P_n'(t_k))^2}.$$

Метод Гаусса обладает высокой точностью при малом числе узлов интегрирования. Если подынтегральная функция сложна и на вычисление ее значений в узлах интегрирования затрачивается много времени, применение формулы Гаусса особенно выгодно.

### Задание

1. Составить схемы алгоритмов интегрирования функций по методам трапеций, Симпсона и Гаусса.
2. Разработать программы интегрирования функций с использованием подпрограмм.
3. Вычислить абсолютные погрешности методов численного интегрирования функций  $\delta = |I^* - I|$ , где  $I^*$  – точное значение интеграла, вычисленное через первообразную;  $I$  – значение интеграла, полученное в результате применения конкретного численного метода.
4. Провести сравнительный анализ методов численного интегрирования.

### Содержание отчета

1. Схемы алгоритмов и программ.
2. Листинг программы.
3. Протоколы результатов решения задачи.
4. Результаты расчетов.
5. Выводы.

Таблица 6.1

№	Подынтегральная функция $f(x)$	Интервал $[a, b]$	Число разбиений $n$	Шаг $h$	Первообразная функция
1	$xe^x \sin x$	[0,1]	50	0,02	$\frac{xe^x(\sin x - \cos x)}{2} + \frac{e^x \cos x - 1}{2}$
2	$\frac{1}{\sqrt{9+x^2}}$	[0,2]	200	0,01	$\ln(x + \sqrt{9+x^2}) - \ln 3$
3	$\frac{1}{\sqrt{1+3x+2x^2}}$	[0,1]	40	0,025	$\frac{1}{\sqrt{2}} \ln \left( \frac{x+0,75 + \sqrt{(x+0,75)^2 - 0,0625}}{0,75+0,5} \right)$
4	$\left(\frac{\ln x}{x}\right)^3$	[1,2]	40	0,025	$(\ln x)^3 + 3(\ln x)^2/2 + \frac{3(\ln x)/2 + 3/4}{2x^2} + \frac{3}{8}$
5	$\frac{x^3}{3+x}$	[1,2]	80	0,0125	$9x - \frac{3x^2}{2} + \frac{x^3}{3} - 27 \ln(3+x) - \frac{47}{6} + 27 \ln 4$
6	$x \operatorname{arctg} x$	[0,2]	50	0,04	$\frac{x^2}{2} \operatorname{arctg} x - \frac{x}{2} + \frac{1}{2} \operatorname{arctg} x$
7	$\frac{1}{x \lg x}$	[2,3]	40	0,025	$2,3026(\ln \ln x - \ln \ln 2)$

## ЧИСЛЕННОЕ РЕШЕНИЕ ОБЫКНОВЕННЫХ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Важное место в теории и практике численных методов решения задач занимают приближенные методы решения задач Коши для обыкновенных дифференциальных уравнений. Многие задачи физики и техники сводятся к нахождению решения именно этой задачи. Обыкновенными дифференциальными уравнениями можно описать задачи движения системы взаимодействующих материальных точек, электрических цепей, сопротивления материалов, систем автоматического управления и регулирования и многое другое. Точное решение дифференциального уравнения можно получить лишь в отдельных случаях, а большинство дифференциальных уравнений могут быть решены только численно. Численное решение задачи состоит в построении таблицы приближенных значений  $y_1, y_2, \dots, y_n$ , решений урав-

нения в точках (узлах) сетки  $x_i = x_0 + ih; i = 1, 2, \dots, n; x_0 = a; x_n = b$ . Величина  $h$  называется шагом интегрирования. Он может быть как постоянным, так и переменным.

Для дифференциального уравнения первого порядка задача Коши ставится следующим образом: требуется найти решение дифференциального уравнения  $y' = f(x, y)$ , удовлетворяющее начальному условию  $y(x_0) = y_0$ . Численные методы не позволяют найти общего решения задачи Коши: они могут дать только какое-то частное решение. Это основной недостаток численных методов. Однако эти методы применимы к очень широким классам уравнений. С появлением современных быстродействующих ЭВМ численные методы решения стали одним из основных способов решения конкретных практических задач для обыкновенных дифференциальных уравнений. Здесь также укажем, что обыкновенное дифференциальное уравнение  $n$ -го порядка, которое можно разрешить относительно старшей производной, легко сводится к системе из  $n$  уравнений первого порядка путем введения  $(n-1)$  новых переменных. Например,  $u'' = g(u', u, x)$  можно записать как систему

$$\left. \begin{aligned} z' &= g(u, z, x), \\ u' &= z \end{aligned} \right\}.$$

### ***Методы Рунге – Кутты***

Одними из более эффективных и часто применяемых на практике численных методов решения задач Коши являются так называемые методы Рунге – Кутты. Эти методы обладают следующими свойствами:

- являются одношаговыми, т.е. для нахождения приближения  $y_{i+1}$  нужна информация только о предыдущей точке  $(x_i, y_i)$ ;
- не требуют вычисления производных от  $f(x, y)$ , необходимо вычисление значений только самой функции  $f(x, y)$ .

Метод Эйлера – простейший численный метод решения задачи Коши. В методе Эйлера величины  $y_i = y(x_i)$  вычисляются по формулам:

$$y_{i+1} = y_i + h(f(x_i, y_i)), \quad (i = 0, 1, 2, \dots, N),$$

$$y_0 = y(x_0), \quad a = x_0 < x_1 < \dots < x_N.$$

Основной недостаток метода Эйлера – систематическое накопление ошибки, которое происходит из-за того, что при вычислении значений на

следующем шаге исходные данные не являются точными и содержат погрешности, зависящие от неточности предыдущих вычислений. Другим недостатком метода является малая точность.

Существуют модифицированные варианты метода Эйлера, которые являются методами Рунге – Кутты второго порядка:

– исправленный метод Эйлера

$$y_{i+1} = y_i + h \cdot \tilde{f}(x_i, y_i, h), \quad y_0 = y(x_0),$$

где  $\tilde{f}(x_i, y_i, h) = \frac{1}{2}[f(x_i, y_i) + f(x_i + h \cdot y_i + h \cdot y_i')]$ ;

– модифицированный метод Эйлера

$$y_{i+1} = y_i + h \cdot \tilde{\tilde{f}}(x_i, y_i, h), \quad y_0 = y(x_0),$$

где  $\tilde{\tilde{f}}(x_i, y_i, h) = \frac{1}{2}[f(x_i, y_i) + f(x_i + \frac{h}{2} \cdot y_i + \frac{h}{2} \cdot y_i')]$ .

Одним из самых употребительных методов численного решения дифференциальных уравнений (систем уравнений) является метод четвертого порядка, называемый просто метод Рунге – Кутты. В этом методе величины  $y_{i+1}$  вычисляются по формуле:

$$y_{i+1} = y_i + h(F(x_i, y_i, h)),$$

где  $F(x_i, y_i, h) = (1/6)(k_{1i} + 2k_{2i} + 2k_{3i} + k_{4i})$ ,

$$k_{1i} = f(x_i, y_i),$$

$$k_{2i} = f(x_i + h/2, y_i + hk_{1i}/2),$$

$$k_{3i} = f(x_i + h/2, y_i + hk_{2i}/2),$$

$$k_{4i} = f(x_i + h, y_i + hk_{3i}).$$

Погрешность метода на одном шаге равна  $Mh^5$ , но поскольку на практике оценить величину  $M$  обычно трудно, при оценке погрешности используют правило Рунге, для этого проводят вычисления сначала с шагом  $h$ , а затем с шагом  $h/2$ . Если  $y_i$  – приближение, вычисленное с шагом  $h$ , а  $y_i$  – с шагом  $h/2$ , то справедлива оценка  $[y_i - y(x_i)] \leq (16/15)[y_i - y_i]$ .

При реализации метода обычно на каждом шаге делают двойной пересчет. Если полученные значения отличаются в пределах допустимой погрешности, то шаг  $h$  удваивают, иначе берут половинный шаг. Для проверки точности можно использовать неравенство  $[y_i - y_i] \leq (E(h)/(b - a))$ .

### **Методы прогноза и коррекции**

В этих методах для вычисления положения новой точки используется информация о нескольких ранее полученных точках. Для этого применяются две формулы, называемые соответственно формулами *прогноза* и *коррекции*. Схемы алгоритмов для всех этих методов примерно одинаковы. Сами методы отличаются лишь формулами решения дифференциального уравнения вида  $y'(x) = f(x, y)$ . Так как в рассматриваемых методах используется информация о нескольких ранее полученных точках, то в отличие от одношаговых методов они не обладают свойством самостартования. Поэтому прежде чем применять метод прогноза и коррекции, приходится вычислять исходные данные с помощью какого-либо одношагового метода. Часто для этого прибегают к методу Рунге – Кутты. Вычисления производят следующим образом. Сначала по формуле прогноза и исходным значениям переменных определяют значение  $y_{n+1}^{(0)}$ . Верхний индекс (0) показывает, что прогнозируемое значение является одним из последовательности значений  $y_{n+1}$ , располагающихся в порядке возрастания точности. По прогнозируемому значению  $y_{n+1}^{(0)}$  с помощью дифференциального уравнения находят производную  $y_{n+1}^{(0)'} = f(x_{n+1}, y_{n+1}^{(0)})$ , которая затем подставляется в формулу коррекции для вычисления уточненного значения  $y_{n+1}^{(j+1)}$ . В свою очередь,  $y_{n+1}^{(j+1)}$  используется для получения более точного значения производной с помощью дифференциального уравнения  $y_{n+1}^{(j+1)'} = f(x_{n+1}, y_{n+1}^{(j+1)})$ .

Если это значение производной недостаточно близко к предыдущему, то оно вводится в формулу коррекции и итерационный процесс продолжается. Если же производная изменяется в допустимых пределах, то значение  $y_{n+1}^{(j+1)}$  используется для вычисления окончательного значения  $y_{n+1}$ . После этого процесс повторяется – делается следующий шаг, на котором вычисляется  $y_{n+2}$ .

### **Метод Милна**

В этом методе на этапе прогноза используется формула Милна:

$$y_{n+1} = y_{n-3} + (4/3)h(2y_n' - y_{n-1}' + 2y_{n-2}') + (28/90)h^5 y^{(5)},$$

а на этапе коррекции – формула Симпсона:

$$y_{n+1} = y_{n-1} + (1/3)h(y_{n+1}' + 4y_n' + y_{n-1}') - (1/90)h^5 y^{(5)}.$$



Последние члены в обеих формулах в действительности в итерационном процессе не используются и служат лишь для оценки ошибки усечения.

Метод Милна относят к методам четвертого порядка точности, так как в нем отбрасываются члены, содержащие  $h$  в пятой и более высокой степенях.

### ***Метод Адамса-Башфорта***

Этот метод также имеет четвертый порядок точности. Используемая в нем формула прогноза получена интегрированием обратной интерполяционной формулы Ньютона и имеет вид:

$$y_{n+1} = y_n + (1/24)h(55y'_n - 59y'_{n-1} + 37y'_{n-2} - 9y'_{n-3}) + (251/270)h^5 y^{(5)}.$$

На этапе коррекции используется формула:

$$y_{n+1} = y_n + (1/24)h(9y'_{n+1} - 19y'_n - 5y'_{n-1} + y'_{n-2}) + (19/720)h^5 y^{(5)}.$$

Расчеты по методу Адамса-Башфорта выполняются так же, как и по методу Милна, однако, в отличие от последнего, ошибка, внесенная на каком-либо шаге, не имеет тенденции к экспоненциальному росту.

### ***Метод Хэмминга***

В методе Хэмминга используются следующие формулы:

$$\text{– прогноза } y_{n+1} = y_{n-3} + \frac{4}{3}h(2y'_n - y'_{n-1} + 2y'_{n-2}) + \frac{28}{90}h^5 y^{(5)};$$

$$\text{– коррекции } y_{n+1} = \frac{1}{8}[9y_n - y_{n-2} + 3h(y'_{n+1} + 2y'_n - y'_{n-1})] - \frac{1}{40}h^5 y^{(5)}.$$

Особенность метода Хэмминга в том, что он позволяет оценивать погрешности, вносимые на стадиях прогноза и коррекции, и устранять их. Благодаря простоте и устойчивости этот метод является одним из наиболее распространенных методов прогноза и коррекции.

### **Задание**

1. Составить схемы алгоритмов решения систем дифференциальных уравнений методами Рунге – Кутты, прогноза и коррекции.

2. Разработать схемы программ решения систем дифференциальных уравнений с использованием подпрограмм.

3. Решить дифференциальные уравнения, приведенные в табл. 7.1. Заданные уравнения предварительно свести к системе уравнений первого порядка. Предусмотреть в программах вычисление точных решений по заданному в таблице алгоритму.

4. Результаты счета оформить в виде графика.

5. Определить близость полученного заданным методом решения к точному значению с помощью оценок:

$$\beta_1 = \max_{i=1, \dots, n} |y_{Ti} - y_{Mi}|, \quad \beta_2 = \frac{\sqrt{\sum_{i=1}^n (y_{Ti} - y_{Mi})^2}}{\sqrt{\sum_{i=1}^n y_{Ti}^2}}.$$

Здесь  $y_{Ti}$  – точное решение;  $y_{Mi}$  – решение, полученное заданным методом;  $n$  – число дискретизации процесса по времени.

6. Провести сравнительный анализ методов решения обыкновенных дифференциальных уравнений.

### Содержание отчета

1. Схемы алгоритмов и программ.
2. Листинг программы.
3. Протоколы результатов решения.
4. Графики сравнительного анализа методов.
5. Результаты расчетов.
6. Выводы.

Таблица 7.1

№ п/п	Дифференциальное уравнение	$y(0), y'(0)$	[a, b]	Шаг $h$	Точное решение
1	$y'' + 2y' + 2y = 2e^{-x} \cos x$	1, 0	[0, 0,5]	0,05	$e^{-x}(\cos x + \sin x + x \sin x)$
2	$y'' + 4y = e^{3x}(13x - 7)$	0, -4	[0, 02]	0,02	$\cos 2x - \sin 2x + e^{3x}(x - 1)$
3	$y'' + 4y' + 4y = 0$	1, -1	[0, 1]	0,1	$(1 + x)e^{-2x}$
4	$y'' - 2y' + y = 5xe^x$	1, 2	[0, 1]	0,1	$e^x + xe^x + 5e^x \frac{x^3}{6}$
5	$y'' - 5y' + 6y = e^x$	0, 0	[0, 0.2]	0,02	$-e^{2x} + 0,5e^{3x} + 0,5e^x$
6	$y'' + y = 1 + e^x$	2.5, 1.5	[0, 1]	0,1	$\cos x + \sin x + 1 + \frac{e^x}{2}$
7	$y'' - 2y' + y = xe^x$	1, 2	[0, 0.5]	0,05	$(1 + x)e^x + (x^3 e^x / 6)$

## **Рекомендательный библиографический список**

1. *Амосов, А. А.* Вычислительные методы для инженеров / А. А. Амосов, Ю. А. Дубинский, Н. В. Копченова. – М. : Высш. шк., 1984. – 584 с.
2. *Волков, Е. А.* Численные методы / Е. А. Волков. – М. : Наука, 1987. – 316 с.
3. *Калиткин, Н. Н.* Численные методы / Н. Н. Калиткин. – М. : Наука, 1987. – 512 с.
4. *Самарский, А. А.* Введение в численные методы / А. А. Самарский. – М. : Наука, 1981. – 286 с.
5. *Тихонов, А. Н.* Вводные лекции по прикладной математике / А. Н. Тихонов, Д. П. Костомаров. – М. : Наука, 1979. – 200 с.
6. *Демидович, Б. П.* Основы вычислительной математики / Б. П. Демидович, И. А. Марон. – М. : Наука, 1966. – 664 с.
7. *Шуп, Т. Е.* Прикладные численные методы в физике и технике / Т. Е. Шуп. – М. : Высш. шк., 1990. – 254 с. – ISBN 5-06001-014-7.
8. *Копченова, Н. В.* Вычислительная математика в примерах и задачах / Н. В. Копченова, И. А. Марон. – М. : Наука, 1982. – 317 с.

## **Оглавление**

Введение.....	3
1. Решение нелинейных уравнений.....	4
2. Решение систем нелинейных уравнений.....	9
3. Решение алгебраических уравнений.....	13
4. Решение систем линейных уравнений.....	16
5. Интерполирование функций.....	20
6. Численное интегрирование функций.....	25
7. Численное решение обыкновенных дифференциальных уравнений.....	29
Рекомендательный библиографический список.....	35

## МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ВЫЧИСЛИТЕЛЬНОЙ ПРАКТИКЕ

Составитель

НОВИКОВА Наталья Александровна

Ответственный за выпуск – зав. кафедрой, профессор И.Н. Егоров

Редактор Л.В. Пукова

Корректор Е.В. Афанасьева

Компьютерная верстка С.В. Павлухиной

ЛР № 020275. Подписано в печать 11.03.05.

Формат 60x84/16. Бумага для множит. техники. Гарнитура Таймс.

Печать на ризографе. Усл. печ. л. 2,09. Уч.-изд. л. 2,21. Тираж 100 экз.

Заказ

Редакционно-издательский комплекс

Владимирского государственного университета.

600000, Владимир, ул. Горького, 87.