

Федеральное агентство по образованию
ГОУ ВПО «Магнитогорский государственный технический
университет им. Г. И. Носова»

Кафедра обработки металлов давлением



ЧИСЛЕННЫЕ МЕТОДЫ

*Методические указания к лабораторным работам
для студентов специальности 150106*

Магнитогорск
2006

Составители: Б.Я. Омельченко
М.И. Румянцев

Численные методы: Методические указания к лабораторным работам для студентов специальности 150106. Магнитогорск: ГОУ ВПО «МГТУ им. Г.И. Носова», 2006. 43 с

Рассмотрены алгоритмы основных численных методов, используемых для решения инженерно-технических задач с применением ПЭВМ. Некоторые из них проиллюстрированы на примерах из обработки металлов давлением

Рецензент М.В. Зарецкий

ВВЕДЕНИЕ

Численные методы – это методы получения решений различных задач не в виде формулы, а в виде одного или нескольких чисел. Их целесообразно применять:

- 1) тогда, когда аналитическое решение невозможно или слишком громоздко;
- 2) при использовании ЭВМ.

Упрощая решение задачи, численные методы, вместе с тем, имеют недостатки:

1. Решения представляются числами и являются частными, так как получены при конкретных значениях исходных данных.

2. Искомые значения определяются с некоторой погрешностью.

Численные методы реализуются с помощью вычислительных алгоритмов и подразделяются на точные и приближенные.

Точными называют методы, которые при отсутствии ошибок вычислений могут дать точный результат **при конечном числе вычислительных и логических операций**.

Приближенные методы дают искомый результат с некоторой погрешностью даже в том случае, если промежуточные вычисления выполняются с абсолютной точностью (без округлений). Среди приближенных численных методов наиболее широко используют **итерационный**. Суть его состоит в том, что решение отыскивается в результате многократного повторения некоторых вычислительных операций. Одно такое повторение называют итерацией. Итерации прекращают тогда, когда погрешность результата станет меньше заранее указанного (принятого) значения. Широкое распространение итерационных методов обусловлено простотой алгоритмов их реализации. Однако при этом возникает проблема сходимости. В некоторых случаях заданная погрешность не может быть достигнута за любое число итераций. Тогда говорят, что сходимость отсутствует.

Погрешность – мера точности найденного значения по сравнению с истинным. Пусть r – результат, а r^* – истинное значение, полученное при вычислениях без округлений и других упрощений. Тогда величину $\Delta r = |r - r^*|$ называют абсолютной, а величину $\delta r = |r - r^*| / r = \Delta r / r$ – относительной погрешностью.

Источниками погрешности приближенного решения являются:

1. Несоответствие математической задачи (математической модели) изучаемому реальному явлению.
2. Погрешность исходных данных (входных параметров).

3. Погрешность метода решения.

4. Ошибки округлений в арифметических и других действиях над числами.

Погрешность в решении, обусловленная первыми двумя источниками, называется **неустранимой**. Эта погрешность может присутствовать, даже если решение поставленной математической задачи найдено точно. Вопрос о том, насколько хорошо описывает математическая модель исследуемое явление, проверяется путем сравнения результатов экспериментов и типичных частных решений при некоторых значениях входных параметров.

Численные методы в большинстве случаев сами по себе являются приближенными, то есть даже при отсутствии ошибок во входных данных и при идеальном выполнении арифметических действий они дают решение задачи с некоторой погрешностью, называемой **погрешностью метода**.

Вопросом, наиболее сложным технически, является учет погрешностей округления в арифметических действиях. Если действий выполняется не много, то **погрешности округления** при ручных вычислениях могут быть учтены с использованием элементарной теории погрешностей. При решении задач на ЭВМ характерны две ситуации. Если количество выполняемых арифметических действий невелико, то обычно ошибки округления не проявляются, так как в ЭВМ числа представляются с 10 и более десятичными значащими цифрами, а окончательный результат редко бывает нужен более чем с пятью десятичными значащими цифрами. Если задача сложная и для ее решения требуется большой объем вычислений (10^6 – 10^7 и выше арифметических действий), то в этом случае с ошибками округления приходится считаться.

Для решения одной и той же задачи могут применяться различные приближенные методы. Чувствительность к ошибкам округления существенно зависит от выбранного численного метода. Малочувствительными к таким ошибкам являются итерационные сходящиеся методы, поскольку возникающие погрешности на следующих итерациях исправляются.

Численный метод может считаться удачно выбранным, если его погрешность в несколько раз меньше неустранимой погрешности, а погрешность, возникающая за счет округлений, называемая **вычислительной погрешностью**, по крайней мере, в несколько раз меньше погрешности метода. Если неустранимая погрешность отсутствует, то погрешность метода должна быть несколько меньше заданной точности решения.

1. ВЫЧИСЛЕНИЕ ЗНАЧЕНИЙ ФУНКЦИЙ

Лабораторная работа № 1

ВЫЧИСЛЕНИЕ ПРОСТЫХ ФУНКЦИЙ

Вычисление значений различных функций наиболее часто используется в вычислительной практике. Все алгоритмические языки, в том числе *Pascal*, *Basic* или *VBA (Visual Basic for Application)*, имеют в своем составе широкий выбор стандартных функций. Для их вычисления необходимо записать указатель и список фактических параметров функции. Для вычисления более сложных функций, в которые входят стандартные функции, необходимо записать соответствующие выражения с учетом правил выбранного алгоритмического языка [1-6]. Вычисление значений функций по заданным аналитическим выражениям (формулам) получило название **прямых вычислений**.

На практике часто приходится решать задачу вычисления таблицы значений функции. В этом случае необходимо организовать циклический вычислительный процесс, задавая в качестве параметров цикла значения аргументов функции.

Пример. Определить длину очага деформации при простой продольной прокатке (рис. 1.1)

$$l = \sqrt{R\Delta h}, \quad (1.1)$$

где R – радиус рабочих валков, мм;

$\Delta h = h_0 - h_1$ – абсолютное обжатие, мм;

h_0 и h_1 – высота полосы до и после прокатки, мм.

Требуется вычислить ряд значений l для нескольких значений радиуса рабочих валков R с определенным шагом и результаты вывести на печать. Алгоритм решения

Рис. 1.1. К расчету длины очага деформации при простой продольной прокатке

этой задачи предполагает прямые вычисления по заданному аналитическому выражению (рис. 1.2).

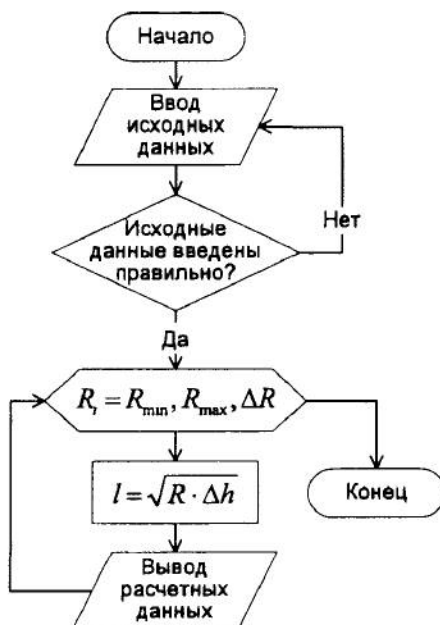


Рис. 1.2. Схема алгоритма расчета нескольких значений функции в цикле

Лабораторная работа № 2

ВЫЧИСЛЕНИЕ ПОЛИНОМА n -Й СТЕПЕНИ

При вычислении значений функций не всегда удастся ограничиться только прямыми вычислениями по формулам. Зачастую для определения значения функции приходится проводить вычисления по некоторому алгоритму, иногда достаточно сложному. При этом вычисляются конечные и бесконечные суммы и произведения, ведется счет по **рекуррентным формулам**. Алгоритмы вычисления значений функции могут носить итерационный характер.

В связи с этим возникает задача такой организации алгоритмов вычисления значений функции, которая гарантировала бы достаточные точность и быстроту проведения вычислений, что особенно важно, если вычисление какой-либо функции должно повторяться многократно. Проиллюстрируем сказанное на примере вычисления значений многочлена степени n при заданном x .

$$P_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0. \quad (1.2)$$

Схема алгоритма вычисления многочлена приведена на рис. 1.3, а. Анализируя этот алгоритм, можно подсчитать, что всего требуется выполнить $n(n+1)/2$ умножений и n сложений.

Алгоритм, представленный на рис. 1.3, а, можно организовать иначе по методу, называемому схемой Горнера. Перепишем многочлен (1.2) в виде

$$P_n(x) = \{ \dots [(a_n x + a_{n-1})x + a_{n-2}]x + \dots + a_1 \}x + a_0. \quad (1.3)$$

По алгоритму, реализующему схему Горнера (рис. 1.3, б), требуется выполнить n умножений и n сложений. Таким образом, число операций умножения, по сравнению с первым алгоритмом, сокращено в $(n+1)/2$ раз, что обеспечивает более высокую точность результата и значительное сокращение времени вычислений.

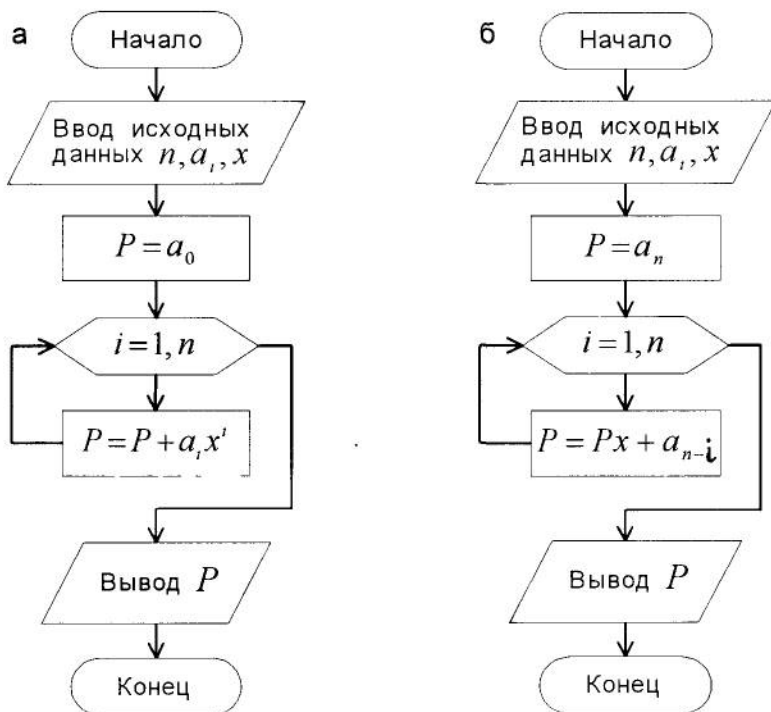


Рис. 1.3 Алгоритмы вычисления многочлена степени n по схемам тривиальной (а) и Горнера (б)

Лабораторная работа № 3

ВЫЧИСЛЕНИЕ ФУНКЦИЙ С ПОМОЩЬЮ РЯДОВ

В вычислительной математике широко используется представление функций степенными, тригонометрическими и другими рядами, в частности, рядами Тейлора и Фурье. В этом случае либо оговаривается число удерживаемых членов ряда (вычисляется конечная сумма членов ряда), либо задается порядок первого отбрасываемого члена (ряд вычисляется с заданной точностью). Получили также распространение вычисления конечных и бесконечных произведений, выполняемые аналогично.

Пример 1 Рассмотрим программную реализацию такого алгоритма. Пусть требуется найти значение многочлена

$$f(x) = 1 + x + x^2/2! + \dots + x^n/n!. \quad (1.4)$$

Приведенный многочлен является частичной суммой разложения $f(x) = e^x$ в ряд Тейлора в окрестности точки $x_0 = 0$.

Записывая последовательно члены этого ряда, имеем $a_0 = 1$, $a_1 = x$ и далее $a_k = a_{k-1}x/k$, где $k = 1, 2, 3, 4, \dots, n$. Эта рекуррентная формула реализуется последовательностью операторов:

```

...
S := 1,0 ;
A := 1,0 ;
for K := 1 to N do
    begin
        A := A * X / K ;
        S := S + A ;
    end ;
...

```

где S – значение суммы ряда; N – заранее заданное число удерживаемых в ряде членов.

В задачах вычисления суммы (произведения) членов бесконечного ряда заранее неизвестно, на каком члене ряда будет достигнута требуемая точность. В этом случае постановка задачи

обычно имеет форму: вычислить сумму $S = \sum_{k=1}^{\infty} a_k$ с точностью до члена ряда, удовлетворяющего неравенству

$$|a_k| \leq \varepsilon. \quad (1.5)$$

Если рассматриваемый ряд является знакочередующимся, то ошибка метода, допускаемая при вычислении S , не превышает ε . В других случаях такая оценка, при которой отбрасываются все члены, удовлетворяющие неравенству (1.5), является ориентировочной и требуются дополнительные соображения для ее принятия. Например, начиная с некоторого номера n , все члены ряда $1 + 1/2 + 1/3 + \dots + 1/n$ удовлетворяют неравенству (1.5), каким бы малым ни было ε , однако этот ряд расходится. Дополнительными соображениями в пользу оценки (1.5) может быть убывание последующего члена ряда по сравнению с предыдущим хотя бы на порядок (в приведенном примере при $|x| < 1$ это условие выполняется, по крайней мере, начиная с $n = 10$).

Приведенная оценка (1.5) является ориентировочной для ошибки метода. Аналогичные требования к оценкам накладываются при вычислении бесконечных произведений.

Пример 2. Вычислить сумму ряда $S = \sum_{k=0}^{\infty} (-1)^k x^{2k} / 2k!$ с точ-

ностью до члена, абсолютное значение которого меньше или равно ε . Вычисляемый ряд представляет собой ряд Тейлора для функции $\cos x$. Схема алгоритма вычисления суммы ряда с заданной точностью приведена на рис. 14. Фрагмент программы, реализующей этот алгоритм, имеет вид:

```

...
S := 1,0;
A := 1,0;
K := 1;
repeat
  A := -A * X / [(2 * K - 1) * 2 * K];
  S := S + A
  K := K + 1
until abs(A) < Eps;
...

```

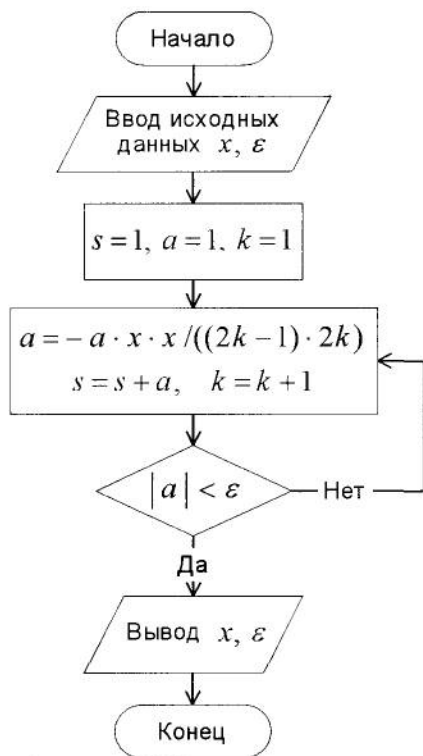


Рис. 1.4. Схема алгоритма вычисления бесконечной суммы с заданной точностью

Таблица сумм некоторых числовых рядов и рядов функций

1. $e = 1 + \frac{1}{1!} + \frac{1}{2!} + \frac{1}{3!} + \dots = \sum_{n=0}^{\infty} \frac{1}{n!}.$
2. $\frac{1}{e} = 1 - \frac{1}{1!} + \frac{1}{2!} - \frac{1}{3!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{1}{n!}.$
3. $\ln 2 = 1 - \frac{1}{2} + \frac{1}{3} - \frac{1}{4} + \dots = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{1}{n}.$
4. $\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots = \sum_{n=0}^{\infty} (-1)^n \frac{1}{2n+1}.$

5. $\frac{\pi^2}{6} = 1 + \frac{1}{2^2} + \frac{1}{3^2} + \frac{1}{4^2} + \dots = \sum_{n=1}^{\infty} \frac{1}{n^2} .$
6. $\frac{\pi^2}{12} = 1 - \frac{1}{2^2} + \frac{1}{3^2} - \frac{1}{4^2} + \dots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{n^2} .$
7. $\frac{\pi^4}{90} = 1 + \frac{1}{2^4} + \frac{1}{3^4} + \frac{1}{4^4} + \dots = \sum_{n=1}^{\infty} \frac{1}{n^4} .$
8. $\frac{7\pi^4}{720} = 1 - \frac{1}{2^4} + \frac{1}{3^4} - \frac{1}{4^4} + \dots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{1}{n^4} .$
9. $\sin x = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^{2n-1}}{(2n-1)!} , (|x| < \infty) .$
10. $\cos x = x - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} , (|x| < \infty) .$
11. $e^x = 1 + \frac{x}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \dots = \sum_{n=0}^{\infty} \frac{x^n}{n!} , (|x| < \infty) .$
12. $\ln x = 2 \left[\frac{x-1}{x+1} + \frac{(x-1)^3}{3(x+1)^3} + \frac{(x-1)^5}{5(x+1)^5} + \dots \right] =$
 $= 2 \sum_{n=0}^{\infty} \frac{(x-1)^{2n+1}}{(2n+1)(x+1)^{2n+1}} , (x > 0) .$
13. $\ln x = (x-1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots =$
 $= \sum_{n=1}^{\infty} (-1)^{n+1} \frac{(x-1)^n}{n} , (0 < x \leq 2) .$
14. $\ln x = \frac{x-1}{x} + \frac{(x-1)^2}{2x^2} + \frac{(x-1)^3}{3x^3} + \dots = \sum_{n=1}^{\infty} \frac{(x-1)^n}{nx^n} , (x > 1/2) .$
15. $\ln(1+x) = x - \frac{x^2}{2} + \frac{x^3}{3} - \frac{x^4}{4} + \dots = \sum_{n=1}^{\infty} (-1)^{n+1} \frac{x^n}{n} , (-1 < x \leq 1) .$

$$16. \ln(1-x) = -(x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4} + \dots) = -\sum_{n=1}^{\infty} \frac{x^n}{n}, \quad (-1 < x \leq 1).$$

$$17. \operatorname{arctg} x = x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}, \quad (|x| \leq 1).$$

$$18. \operatorname{arcctg} x = \frac{\pi}{2} - (x - \frac{x^3}{3} + \frac{x^5}{5} - \frac{x^7}{7} + \dots) = \frac{\pi}{2} + \sum_{n=0}^{\infty} (-1)^{n+1} \frac{x^{2n+1}}{2n+1}, \quad (|x| \leq 1).$$

$$19. \operatorname{sh} x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots = \sum_{n=1}^{\infty} \frac{x^{2n-1}}{(2n-1)!}, \quad (|x| < \infty).$$

$$20. \operatorname{ch} x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}, \quad (|x| < \infty).$$

2. ПРИБЛИЖЕНИЕ ФУНКЦИЙ

Если функция $y = f(x)$ задана, то тем самым любому допустимому значению x сопоставляется значение y . Однако чаще всего удастся получить небольшое число значений функции. Вместе с тем для расчетов функции желательно иметь достаточно простую аналитическую зависимость. Тогда функцию заменяют приближенной формулой $\varphi(x)$, близкой к функции $f(x)$ в некотором смысле и удобной для вычислений.

Различают два основных способа выбора приближения функций: **интерполяцию** и **аппроксимацию**.

Пусть функция $f(x)$ задана таблицей ее значений $y = y_0, y_1, y_2, \dots, y_n$ в точках $x = x_0, x_1, x_2, \dots, x_n$, называемых узлами интерполяции. Задача интерполяции состоит в выборе такой функции $\varphi(x)$, которая в узлах x_i принимала бы те же значения, что и $f(x)$, то есть $\varphi(x_i) = y_i, i = 1 \dots n$.

Как правило, в качестве интерполирующих функций выбирают многочлены. Поскольку число узлов $n+1$, то степень интерполяционного многочлена равна n .

Во многих случаях, особенно когда значения функции $f(x)$ получены экспериментально и содержат погрешности, нет необхо-

димости точного совпадения значений (x) и $f(x)$ в узловых точках. Более того, для практики можно пользоваться многочленом степени m , причем $m < n$, или аналитической зависимостью иного вида, но график ее должен проходить достаточно близко от точек x_i, y_i на всем протяжении интервала изменения x . Процедура построения функции, исходя из заранее определенного понятия близости, называется аппроксимацией.

Лабораторная работа № 4

ИНТЕРПОЛЯЦИЯ ФУНКЦИЙ С ПОМОЩЬЮ МНОГОЧЛЕНОВ

Интерполяционная формула Ньютона. Требуется найти многочлен $P_n(x)$ степени n , принимающий в заданных точках $x = x_0, x_1, \dots, x_n$ те же значения, что и данная функция $f(x)$. Решим задачу с помощью интерполяционной формулы Ньютона.

Введем понятие разделенных разностей. Отношения

$$A_{1i} = A(x_i, x_{i+1}) = [f(x_{i+1}) - f(x_i)] / (x_{i+1} - x_i), \quad i = 0, \dots, n-1$$

называются разделенными разностями первого порядка. Отношения

$$\begin{aligned} A_{2i} = A(x_i, x_{i+1}, x_{i+2}) &= [A(x_{i+1}, x_{i+2}) - A(x_i, x_{i+1})] / (x_{i+2} - x_i) = \\ &= (A_{1,i+1} - A_{1i}) / (x_{i+2} - x_i) \end{aligned}$$

называются разделенными разностями второго порядка. Следующие разделенные разности вычисляются по формуле

$$A_{mi} = (A_{m-1,i+1} - A_{m-1,i}) / (x_{i+m} - x_i).$$

Рассмотрим многочлен второй степени

$$\begin{aligned} P_2(x) &= f(x_0) + A_{10}(x_0, x_1)(x - x_0) + \\ &+ A_{20}(x_0, x_1, x_2)(x - x_0)(x - x_1). \end{aligned} \quad (2.1)$$

Имеем: $P_2(x_0) = f(x_0)$;

$$P_2(x_1) = f(x_0) + A_{10}(x_0, x_1)(x_1 - x_0) = f(x_1);$$

$$\begin{aligned} P_2(x_2) &= f(x_0) + A_{10}(x_0, x_1)(x_2 - x_0) + \\ &+ A_{20}(x_0, x_1, x_2)(x_2 - x_0)(x_2 - x_1) = f(x_2). \end{aligned}$$

Таким образом, многочлен (2.1), построенный с помощью первых (т.е. начиная с $i = 0$) разделенных разностей первого и второго порядка, в точках x_0, x_1, x_2 принимает те же значения, что и функция $f(x)$.

Имеет место следующее утверждение: многочлен степени n

$$\begin{aligned} P_n(x) = & f(x_0) + A_{10}(x_0, x_1)(x - x_0) + \\ & + A_{20}(x_0, x_1, x_2)(x - x_0)(x - x_1) + \dots + \\ & + A_{n0}(x_0, x_1, \dots, x_n)(x - x_0)(x - x_1) \dots (x - x_{n-1}) \end{aligned} \quad (2.2)$$

в точках x_0, x_1, \dots, x_n принимает те же значения, что и функция $f(x)$. Выражение (2.2) называется интерполяционным многочленом Ньютона

Для функции, заданной таблично при равноотстоящих узлах, приведем схему алгоритма вычисления значений интерполяционного многочлена Ньютона (рис. 2.1). Исходными данными для алгоритма являются

x – начальный узел интерполяции;

h – шаг интерполяции;

n – число узлов;

z – значение аргумента, для которого вычисляется значение многочлена;

k_y – вспомогательная переменная $k_y = 0$ при вычислении первого значения интерполяционного многочлена; $k_y = 1$ при последующих вычислениях);

$y(n)$ – массив значений функции (при $k_y = 0$) или массив разделенных разностей (при $k_y = 1$);

P – значение интерполяционного многочлена.

Для равноотстоящих узлов разделенные разности вычисляются по формуле

$$A_m = (A_{m-1, i+1} - A_{m-1, i}) / (mh),$$

где $m = 1, n - 1$.

Для вычисления интерполяционного многочлена выражение (2.2) преобразуется к виду

$$P_n(x) = (\dots((A_{n0}(x - x_{n-1}) + A_{n-1,0})(x - x_{n-2}) + A_{n-2,0})\dots) \times (x - x_0) + A_0.$$

Здесь $A_0 = y_1$.

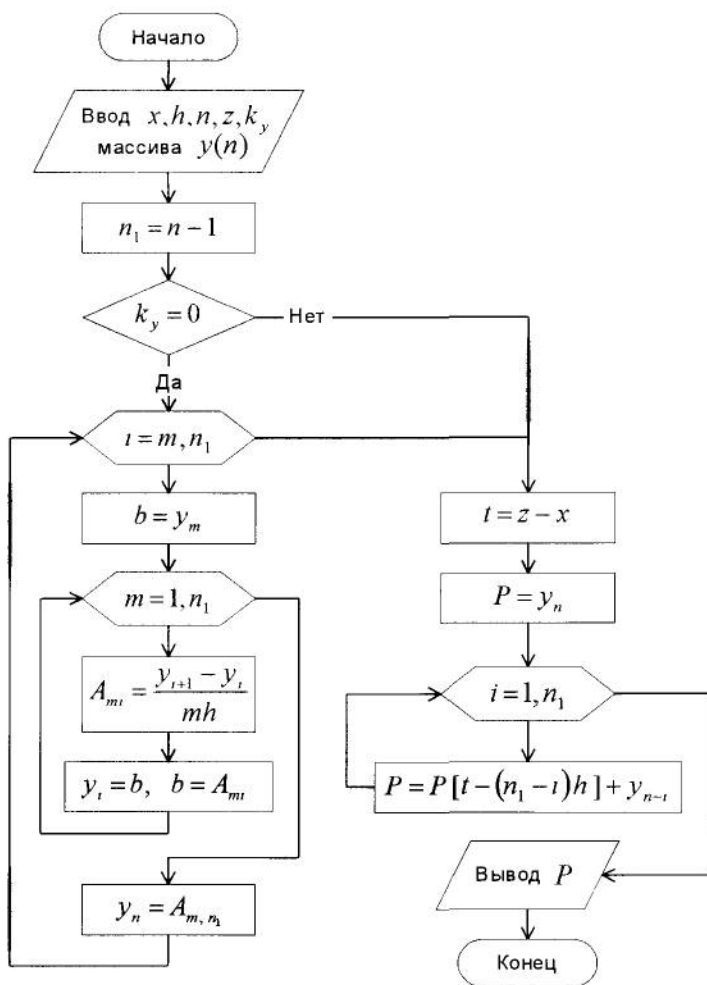


Рис. 2.1 Схема алгоритма вычисления значений интерполяционного многочлена Ньютона

Пример Функция задана таблицей

x	1,1	1,6	2,1	2,6	3,1	3,6	4,1	4,6
y	1,029	1,389	1,649	1,800	1,852	1,822	1,739	1,632

Необходимо найти ее приближенное значение с помощью интерполяционного многочлена Ньютона при $x_1 = 1,3$ и $x_2 = 4,0$

Составление интерполяционной формулы Ньютона для большого числа точек связано с тем, что по мере продвижения от начальной точки накапливаются ошибки, обусловленные вычислением разделенных разностей. Ввиду потери точности использование разделенных разностей больших порядков считается нецелесообразным.

Приведем иной прием составления интерполяционного многочлена, основанный на одновременном введении в вычисления всех точек.

Интерполяционный многочлен Лагранжа Введем функции

$$L_i^{(n)}(x) = \frac{(x-x_0)(x-x_1) \dots (x-x_{i-1})(x-x_{i+1}) \dots (x-x_n)}{(x_i-x_0)(x_i-x_1) \dots (x_i-x_{i-1})(x_i-x_{i+1}) \dots (x_i-x_n)},$$

называемые коэффициентами Лагранжа и удовлетворяющие условиям

$$L_i^{(n)}(x_i) = 1, \quad L_i^{(n)}(x_j) = 0 \quad \text{при } i \neq j$$

Интерполяционным многочленом Лагранжа называется многочлен вида

$$L_n(x) = \sum_{i=0}^n y_i L_i^{(n)}(x)$$

Интерполяционный многочлен Лагранжа применяется в тех же случаях, что и формула Ньютона (2.2)

Лабораторная работа № 5

АППРОКСИМАЦИЯ ФУНКЦИЙ. МЕТОД НАИМЕНЬШИХ КВАДРАТОВ

Задача интерполяции состояла в построении многочлена $P_n(x)$ степени n , значения которого в $n+1$ точках интерполяции совпадают со значениями данной функции $f(x)$

$$P_n(x_i) = f(x_i) \tag{2.3}$$

С увеличением n трудности построения интерполяционного многочлена существенно возрастают. Однако необходимость в точном выполнении равенства (2.3) возникает не всегда и часто достаточно иметь многочлен значительно меньшей степени, например, линейную функцию или квадратный трехчлен, но отражающий представляющие прикладной интерес основные закономерности исследуемой зависимости. Поэтому естественно поставить задачу об определении многочлена более низкой степени m ($m < n$)

$$P_m(x) = a_m x^m + \dots + a_1 x + a_0, \quad (2.4)$$

«расстояние» между которым и функцией $f(x)$ в некотором смысле минимально. Такой многочлен называется аппроксимирующим данную функцию.

Аппроксимирующий многочлен сглаживает локальные особенности заданной экспериментальной таблицы и отражает общее поведение функции $f(x)$ вдоль всего интервала ее изменения.

Изложим метод **наименьших квадратов**, решающий поставленную задачу [3]. Пусть дана таблица значений функции $y = f(x)$.

$$\{x_i, y_i\}, i = 1, \dots, n.$$

Аппроксимирующая функция разыскивается в виде

$$Y = (x, a_1, \dots, a_m), \quad (2.5)$$

где a_1, \dots, a_m – подлежащие определению параметры, число которых m значительно меньше n .

Вычислим значения функции (2.5) при $x = x_i$.

$$Y_i = \varphi(x_i, a_1, \dots, a_m), i = 1, \dots, n$$

Между исследуемой и аппроксимирующей функциями имеется различие, которое оценим с помощью суммы квадратов разностей их значений

$$S = S(a_1, \dots, a_m) = \sum_{i=1}^n (Y_i - y_i)^2 = \sum [\varphi(x_i, a_1, \dots, a_m) - y_i]^2. \quad (2.6)$$

Параметры a_1, \dots, a_m найдем из условия минимума ошибки аппроксимации (2.6).

$$\frac{\partial S}{\partial a_k} = 2 \sum_{i=1}^n [\varphi(x_i, a_1, \dots, a_m) - y_i] \frac{\partial}{\partial a_k} \varphi(x_i, a_1, \dots, a_m) = 0. \quad (2.7)$$

В результате получаем систему m уравнений для определения m неизвестных параметров.

Аппроксимация многочленом. Пусть аппроксимирующая функция (2.5) имеет вид

$$Y = a_1 x + a_0. \quad (2.8)$$

Тогда, согласно выражению (2.6),

$$S = \sum_{i=1}^n (a_1 x_i + a_0 - y_i)^2.$$

Уравнения (2.7) для определения параметров примут вид:

$$\frac{\partial S}{\partial a_1} = 2 \sum_{i=1}^n (a_1 x_i + a_0 - y_i) x_i = 0,$$

$$\frac{\partial S}{\partial a_0} = 2 \sum_{i=1}^n (a_1 x_i + a_0 - y_i) = 0,$$

или

$$a_1 \sum_{i=1}^n x_i^2 + a_0 \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i \quad (2.9)$$

$$a_1 \sum_{i=1}^n x_i + n a_0 = \sum_{i=1}^n y_i.$$

Коэффициенты a_0, a_1 определяются однозначно:

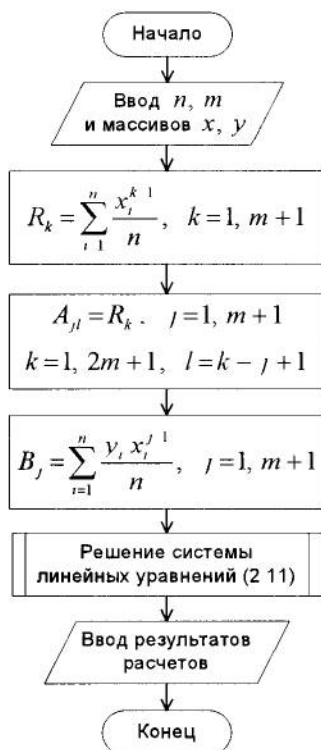
$$a_1 = \frac{\sum x_i \sum y_i - n \sum x_i y_i}{(\sum x_i)^2 - n \sum x_i^2}, \quad (2.10)$$

$$a_0 = \frac{1}{n(\sum y_i - a_1 \sum x_i)}.$$

Если аппроксимирующая функция имеет вид уравнения (2.4), то коэффициенты a_m, \dots, a_1, a_0 определяются из системы линейных уравнений, аналогичных уравнениям (2.9):

$$\begin{aligned} R_0 a_0 + R_1 a_1 + \dots + R_m a_m &= B_0, \\ R_1 a_0 + R_2 a_1 + \dots + R_{m+1} a_m &= B_1; \\ R_m a_0 + R_{m+1} a_1 + \dots + R_{2m} a_m &= B_m, \end{aligned} \quad (2.11)$$

где $R_k = \sum_{i=1}^n x_i^k / n$, $k = 0, \dots, 2m$; $B_j = \sum_{i=1}^n y_i x_i^j$, $j = 0, \dots, m$. (2.12)



Погрешность среднеквадратической аппроксимации функции определяется выражением

$$\Delta = \left[\sum_{i=0}^n (f(x_i) - P_n(x_i))^2 / (n+1) \right]^{1/2}$$

В алгоритме аппроксимации многочленом методом наименьших квадратов (рис. 2.2) приняты следующие обозначения:

n – количество точек, в которых известны значения функции; m – степень аппроксимирующего многочлена; X , Y – массивы табличных значений аргумента и функции; A – массив коэффициентов системы линейных уравнений (2.11); R , B – массивы коэффициентов и свободных членов (2.12).

Рис. 2.2 Схема алгоритма аппроксимации функции методом наименьших квадратов

3. РЕШЕНИЕ НЕЛИНЕЙНЫХ УРАВНЕНИЙ

Нелинейным называется уравнение вида

$$f(x) = 0, \quad (3.1)$$

где $f(x)$ – алгебраическая или трансцендентная функция переменного x , определенная и непрерывная на некотором интервале $[a, b]$. Всякое значение $x \in [a, b]$, при котором функция обращается в нуль, называется корнем уравнения (3.1).

Уравнение (3.1) называется **алгебраическим**, если его левая часть имеет вид многочлена

$$a_n x^n + a_{n-1} x^{n-1} + \dots + a_2 x^2 + a_1 x + a_0 = 0. \quad (3.2)$$

Это уравнение имеет n действительных или комплексных корней.

Если левая часть уравнения (3.1) является неалгебраической функцией, например, логарифмической, показательной, тригонометрической и т.д., то уравнение (3.1) называется **трансцендентным**.

Предполагается, что внутри интервала $[a, b]$ существует такой интервал $[x_1, x_2]$, на котором при $a \leq x_1 < x_2 \leq b$ функция $f(x)$ непрерывна и меняет знак, то есть $f(x_1) \cdot f(x_2) < 0$, а ее производная $f'(x)$ знак сохраняет. Согласно первому условию корень в интервале $[x_1, x_2]$ существует и по второму условию, равносильному требованию монотонности $f(x)$ при изменении x , этот корень в интервале $[x_1, x_2]$ единственный. Такой интервал называется интервалом изоляции корня.

Задача численного нахождения корней уравнения (3.1) обычно разбивается на 2 этапа: отделение корней уравнения $f(x) = 0$, уточнение корней уравнения.

На первом этапе определяются интервалы изоляции корней, а на втором этапе уточняются эти корни с помощью численных методов для достижения заданной степени точности нахождения корней уравнения. Можно назвать следующие **итерационные численные** методы

- метод половинного деления;

- метод хорд;
- метод касательных (метод Ньютона);
- метод простой итерации и др.

Лабораторная работа № 6

МЕТОД ПОЛОВИННОГО ДЕЛЕНИЯ

Рассмотрим **метод половинного деления**, который является одним из наиболее простых методов и состоит в следующем.

Интервал изоляции корня уравнения делится пополам (рис 3 1). Если $f((x_1 + x_2)/2) \neq 0$, то выбирается одна из половин $[x_1, (x_1 + x_2)/2]$ или $[(x_1 + x_2)/2, x_2]$, на концах которой $f(x)$ принимает значения разных знаков. Далее эта операция повторяется.

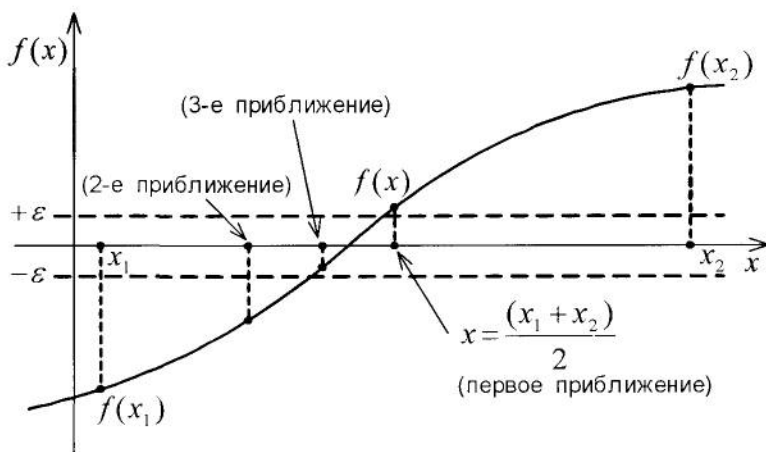


Рис 3 1 Графическая иллюстрация метода половинного деления

Итерационный процесс приводит к тому, что после n итераций длина суженного интервала Δx_n становится в два раза меньше первоначальной длины $|x_2 - x_1|$ и равной $(x_2 - x_1)/2^n$. Следовательно, для выполнения условия $\Delta x_n \leq \varepsilon$, где ε – задаваемая точность вычисления корня, требуется выполнить n итераций.

$$n \approx \ln((x_2 - x_1)/\varepsilon) / \ln 2 \quad (3.3)$$

На практике в качестве критерия окончания итерационного процесса вычисления корня уравнения (3.1) часто используют условие выполнения неравенства $|f(x)| \leq \varepsilon$. Критерий достаточно эффективен, если вблизи искомого значения x имеет место $|f'(x)| > 1$.

Схема алгоритма уточнения корня уравнения $f(x) = 0$ в интервале $[x_1, x_2]$ с точностью ε методом половинного деления приведена на рис. 3.2.

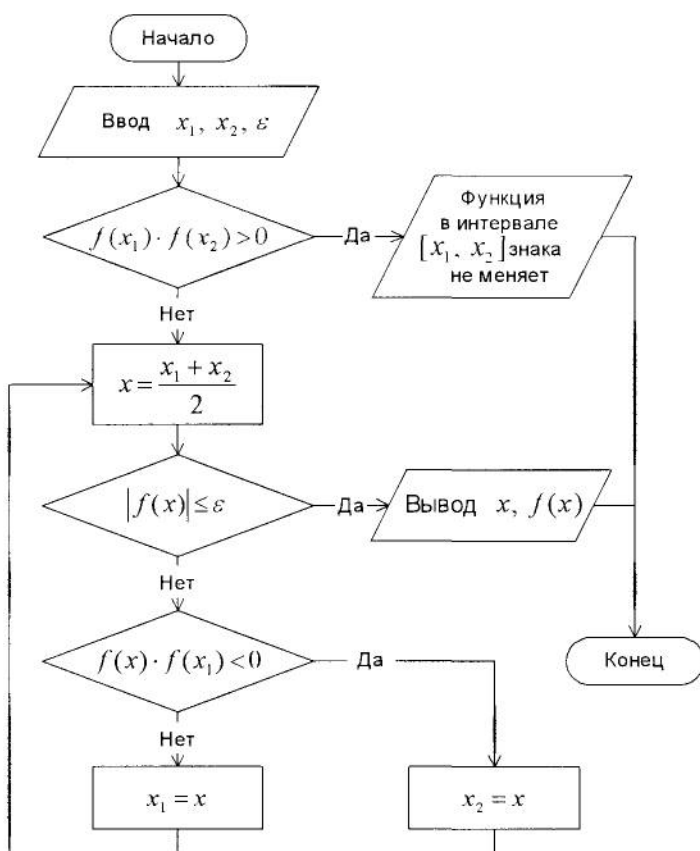


Рис. 3.2. Схема алгоритма метода половинного деления

Пример. Рассмотрим упругопластическую деформацию толстостенной трубы под действием внутреннего и наружного давле-

ний в условиях плоского деформированного состояния (рис. 3.3). Для нахождения радиуса R_c необходимо решить трансцендентное уравнение [7]

$$(p_a - p_b)/\tau_T - 2\ln(R_c/R_a) = 1 - (R_c/R_b)^2, \quad (3.4)$$

где p_a и p_b – внутреннее и внешнее давление;

τ_T – предел текучести на сдвиг материала трубы;

R_a и R_b – внутренний и внешний радиусы трубы;

R_c – граница между упругой и пластической зонами деформациями.

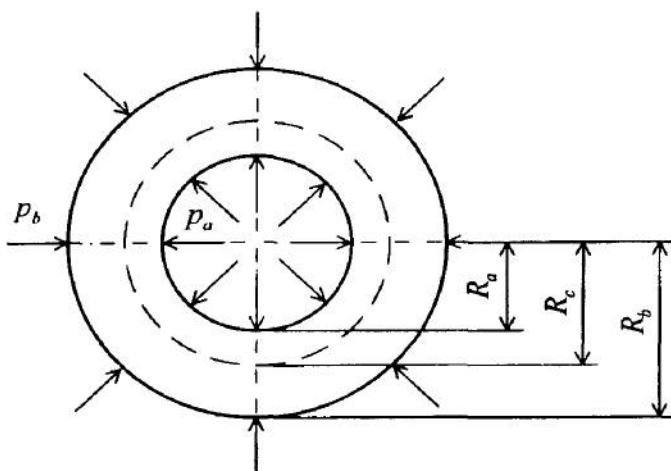


Рис. 3.3. К расчету границы между упругой и пластической зонами деформации

Лабораторная работа № 7

МЕТОД ПРОСТОЙ ИТЕРАЦИИ

Рассмотрим расчет среднего удельного давления при холодной прокатке (по А.И. Целикову) с учетом упругого сплющивания рабочих валков [8, 9].

$$p_{cp} = 1,15 \sigma_s n'_\sigma n'''_\sigma, \quad (3.5)$$

где σ_s – усредненный по очагу деформации предел текучести материала полосы при линейном напряженном состоянии

$$\sigma_s = \frac{\sigma_{s0} + \sigma_{s1}}{2}, \quad (3.6)$$

σ_{s0} и σ_{s1} – предел текучести полосы до и после клетки (прохода);

n'_σ – коэффициент, учитывающий влияние внешнего трения и сложной формы очага деформации

$$n'_\sigma = \frac{2h_n}{\Delta h(\delta - 1)} \left[\left(\frac{h_n}{h_1} \right)^\delta - 1 \right], \quad (3.7)$$

$\Delta h = h_0 - h_1$ – абсолютное обжатие,

h_0 и h_1 – высота полосы до и после клетки (прохода),

h_n – высота полосы в нейтральном сечении

$$h_n = h_1 \left[\frac{1 + \sqrt{1 + (\delta^2 - 1)(h_0/h_1)^\delta}}{\delta + 1} \right]^{1/\delta}, \quad (3.8)$$

δ – коэффициент напряженного состояния

$$\delta = \frac{2fl_c}{\Delta h}, \quad (3.9)$$

f – коэффициент контактного трения,

l_c и l – длина очага деформации с учетом и без учета упругого сплющивания рабочих валков

$$l_c = \sqrt{R\Delta h + v_0^2} + x_0, \quad (3.10)$$

$$l = \sqrt{R\Delta h}, \quad (3.11)$$

$$x_0 = \frac{8(1 - \nu_p^2)}{\pi E_p} R p_{cp}, \quad (3.12)$$

R – радиус рабочих валков;

ν_p и E_p – коэффициент Пуассона и модуль Юнга 1-го рода материала рабочих валков;

n_{σ}''' – коэффициент, учитывающий влияние натяжения.

$$n_{\sigma}''' = 1 - \frac{\sigma_0 + \sigma_1}{2,3\sigma_s} = 1 - \frac{K_{\sigma 0}\sigma_{s0} + K_{\sigma 1}\sigma_{s1}}{2,3\sigma_s}; \quad (3.13)$$

σ_0 и σ_1 – заднее и переднее средние удельные натяжения полосы;

$K_{\sigma 0}$ и $K_{\sigma 1}$ – коэффициенты, показывающие, какую долю от соответствующих пределов текучести составляют заднее и переднее натяжения полосы.

Из представленных уравнений видно, что p_{cp} зависит от l_c (см. уравнения (3.5), (3.7)–(3.9)), а l_c , в свою очередь, зависит от p_{cp} (см. уравнения (3.10) и (3.11)), то есть

$$p_{cp} = f(l_c) \text{ и } l_c = (p_{cp}). \quad (*)$$

Для решения этой системы на ЭВМ наиболее удобно использовать метод последовательных приближений – **метод простой итерации**. Сущность его заключается в следующем. Сначала находят длину очага деформации без учета сплющивания по формуле (3.11), затем определяют среднее удельное давление по формуле (3.5). После этого находят в первом приближении длину очага деформации с учетом сплющивания согласно формулам (3.10), (3.12) и определяют уточненное значение среднего удельного давления. Далее находят длину очага деформации с учетом сплющивания во втором приближении и т.д., пока найденные значения не будут удовлетворять уравнениям (*). Расчет обычно ведут до достижения заданной точности ε .

$$\left| 1 - \frac{l'_c}{l_c} \right| \leq \varepsilon, \quad (3.14)$$

где l'_c и l_c – предыдущее и последующее значения длины очага деформации с учетом упругого сплющивания рабочих валков.

Графически такой итерационный процесс можно представить следующим образом (рис. 3 4, а). Однако следует иметь в виду, что при некоторых параметрах процесса прокатки система уравнений (*) может и не иметь решения. В этом случае итерационный цикл никогда не сойдется (рис 3 4, б), то есть процесс вычислений будет расходящимся и приведет к тому, что значения переменных превысят допустимые значения. В качестве критерия сходимости итерационного цикла можно использовать, например, следующее условие.

$$l_c < 10l, \quad (3.15)$$

Прокатка при заданных параметрах процесса будет возможна только тогда, когда будет выполняться неравенство (3.15).

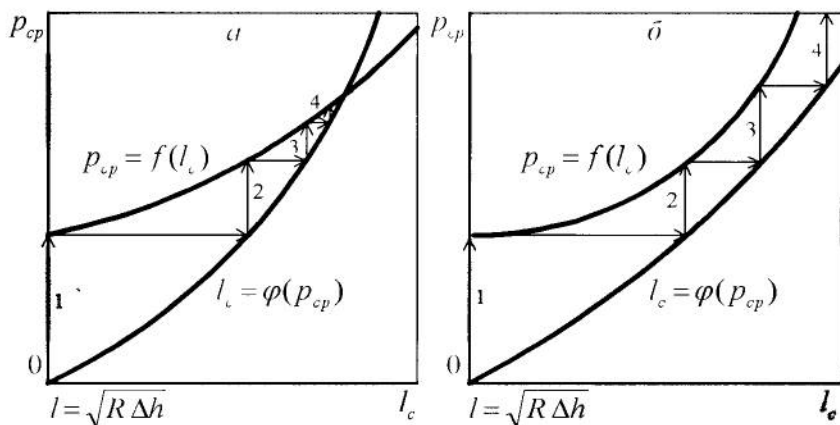


Рис. 3.4. Сходящийся (а) и расходящийся (б) итерационные циклы при определении l_c и p_{cp} ; цифры у вертикальных отрезков – порядковый номер итерации

Алгоритм расчета среднего удельного давления с учетом упругого сплющивания рабочих валков представлен на рис. 3.5.

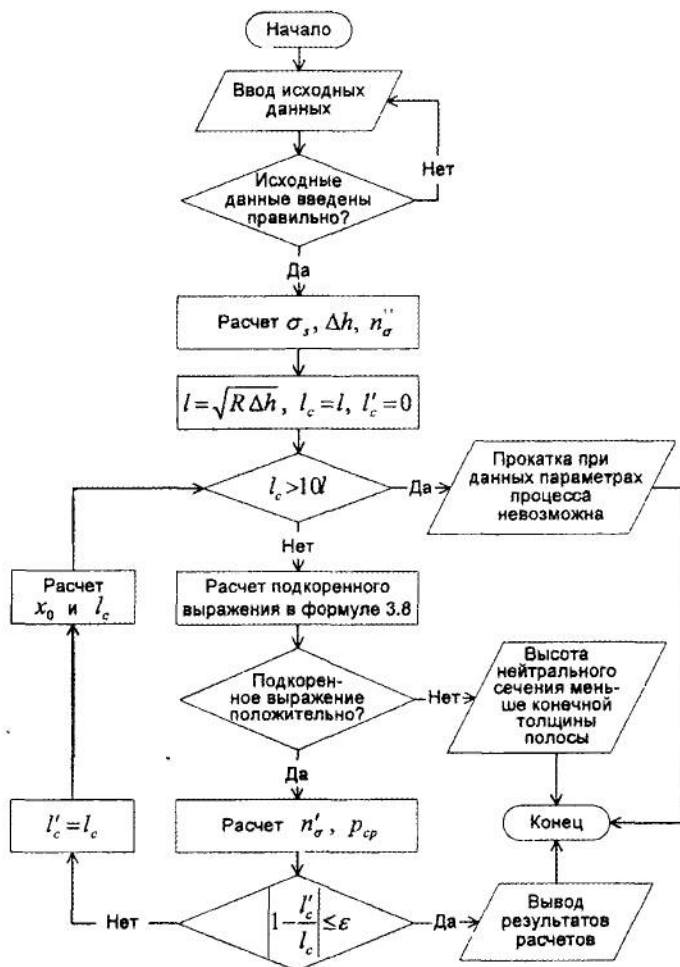


Рис. 3.5. Схема алгоритма расчета среднего удельного давления с учетом упругого сплющивания рабочих валков

4. РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ

Лабораторная работа № 8

МЕТОД ГАУССА

Решение систем линейных алгебраических уравнений является одной из задач, наиболее часто встречающихся на практике как

(4.2) на a_{22} , умножаем полученное уравнение на a'_{32}, \dots, a'_{n2} , вычитаем его из последующих уравнений. Получаем

$$\begin{aligned} x_1 + a'_{12}x_2 + a'_{13}x_3 + \dots + a'_{1n}x_n &= b'_1; \\ x_2 + a''_{23}x_3 + \dots + a''_{2n}x_n &= b''_2; \\ a'''_{33}x_3 + \dots + a'''_{3n}x_n &= b'''_3; \\ &\vdots \\ x_n &= b^{(n)}_n. \end{aligned}$$

Повторяя описанные преобразования вдоль всей системы (4.1), приводим ее к виду

$$\begin{aligned} x_1 + a'_{12} + \dots + a'_{1n} x_n &= b'_n; \\ x_2 + \dots + a''_{2n} x_n &= b''_n; \\ &\dots \dots \dots \\ x_n &= b^{(n)}_n. \end{aligned} \quad (4.3)$$

На этом заканчивается **прямой ход** метода Гаусса. Имеем $x_n = h_n^{(n)}$. Выполнив обратную прогонку для всех уравнений системы (4.3), получим $x_{n-1}, x_{n-2}, \dots, x_2, x_1$. Эту часть процесса вычислений называют **обратным ходом** метода Гаусса. При этом следует отметить следующее.

Операции прямого хода проводятся только с коэффициентом при неизвестных, а не с самими неизвестными. Поэтому в память ЭВМ следует вводить только матрицу коэффициентов A , составленную из коэффициентов при неизвестных и свободных членов $b_i = a_{i, n+1}$.

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & a_{1,n+1} \\ a_{21} & a_{22} & \dots & a_{2n} & a_{2,n+1} \\ \dots & \dots & \dots & \dots & \dots \\ a_{n1} & a_{n2} & \dots & a_{nn} & a_{n,n+1} \end{pmatrix}.$$

В целях экономии памяти ЭВМ результаты преобразования коэффициентов матрицы A можно размещать в тех же ячейках памяти согласно индексным переменным.

Алгоритм метода Гаусса состоит из следующих основных этапов:

1. Первая строка матрицы делится на a_{11} , затем умножается на a_{k1} ($k = 2, 3, \dots, n$) и вычитается из k -й строки (последовательно из $2, 3, \dots, n$). Элементы первого столбца матрицы (с a_{21} до a_{n1}) становятся нулями

$$A^{(1)} = \begin{pmatrix} 1 & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} & a_{2,n+1}^{(1)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} & a_{n,n+1}^{(1)} \end{pmatrix}$$

2. Вторая строка преобразованной матрицы A делится на $a_{22}^{(1)}$, а затем умножается на $a_{k2}^{(1)}$ и вычитается из всех строк с $k = 3, 4, \dots, n$.

3. Действия, аналогичные перечисленным в пунктах 1, 2, повторяются до тех пор, пока подобная процедура не будет проделана с $(n-1)$ -й строкой матрицы, при этом матрица будет приведена к виду

$$A^{(n-1)} = \begin{pmatrix} 1 & a_{12}^{(1)} & \dots & a_{1n}^{(1)} & a_{1,n+1}^{(1)} \\ 0 & 1 & \dots & a_{2n}^{(2)} & a_{2,n+1}^{(2)} \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & a_{n-1,n}^{(n-2)} & a_{n-1,n+1}^{(n-2)} \\ 0 & 0 & \dots & a_{nn}^{(n-1)} & a_{n,n+1}^{(n-1)} \end{pmatrix}.$$

4. Элементы последней строки полученной матрицы позволяют вычислить значение

$$x_n = a_{n,n+1}^{(n-1)} / a_{nn}^{(n-1)}.$$

5. Значение корня x_n используется для отыскания x_{n-1} при подстановке в $(n-1)$ -ю строку треугольной матрицы $A^{(n-1)}$, а затем последовательно вычисляются корни x_{n-2}, \dots, x_1 по формуле

$$x_k = a_{k,n-1}^{(k-1)} - \sum_{j=k+1}^n a_{kj}^{(k-1)} x_j, \quad k = n-1, \dots, 1.$$

Метод Гаусса позволяет построить экономичный алгоритм, поскольку элементы матрицы преобразуются по достаточно простым формулам и значения преобразовательных элементов представляют собой текущие значения тех же переменных, что и значения элементов исходной матрицы.

Алгоритм решения системы n линейных алгебраических уравнений методом Гаусса приведен на рис. 4.1. Алгоритм метода Гаусса достаточно прост и экономичен по числу арифметических операций, однако, для его применения необходимо, чтобы все ведущие элементы $a_{ij}^{(i-1)}$ ($i = 2, 3, \dots, n$) были отличны от нуля. Близость ведущих элементов к нулю может привести к большой потере точности решения.

В связи с этим были предложены различные варианты метода Гаусса с выбором главных элементов, например, схема с выбором главных элементов по всей матрице. В данном случае перед проведением операции исключения из элементов преобразуемой матрицы на каждом i -м этапе ($i = 1, 2, \dots, n-1$) выбирается наибольший по модулю элемент, называемый главным. Стоящее при нем неизвестное исключается по описанному выше правилу. Для удобств вычислений перед исключением этого неизвестного делают перестановку уравнений и неизвестных так, чтобы главный элемент занял левый верхний угол преобразуемой матрицы. Если на i -м шаге наибольший элемент выбирается только среди элементов i -го столбца (строки), то такая схема называется схемой с выбором главного элемента по столбцу (строке). Для решения произвольных систем на ЭВМ целесообразно использовать метод Гаусса с тем или иным выбором главного элемента.

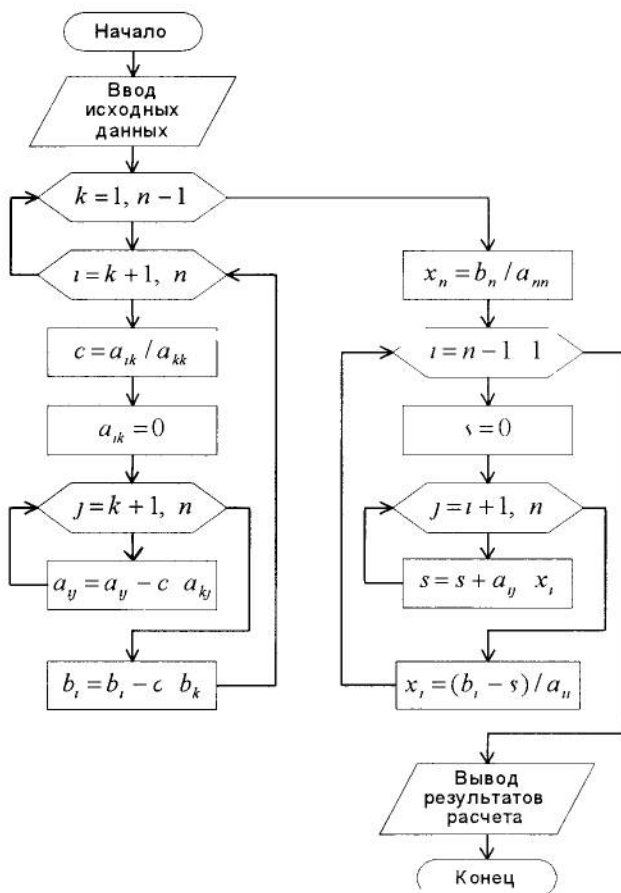


Рис 4 1 Блок-схема алгоритма решения системы линейных уравнений методом Гаусса

5. ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ ФУНКЦИЙ И ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Лабораторная работа № 9

ИНТЕГРИРОВАНИЕ ФУНКЦИЙ

Формула Ньютона-Лейбница $\int_a^b f(x)dx = F(b) - F(a)$ для вы-

числения определенного интеграла предполагает известной пер-

вообразную $F(x)$ подынтегральной функции $f(x)$. Однако выразить первообразную через элементарные функции удастся далеко не всегда, причем часто ее вид плохо приспособлен для дальнейших вычислений. В этом случае прибегают к численному интегрированию, наиболее распространенными методами которого являются **метод трапеций** и **метод Симпсона**.

Пусть на интервале (a, b) функция $f(x)$ представлена интерполяционным многочленом Лагранжа степени n

$$L_n(x) = \sum_{i=0}^n y_i L'_n(x)$$

с узлами в точках $x_0 = a$; $x_i = x_0 + ih$; $h = (b-a)/n$; $i = 1, \dots, n$ и значениями в этих точках $y_i = f(x_i)$. Тогда

$$\int_a^b f(x) dx = \int_a^b L_n(x) dx + R_n = \sum_{i=0}^n A_i y_i + R_n, \quad (5.1)$$

где R_n – ошибка квадратурной формулы; $A_i = \int_a^b L'_n(x) dx$ – постоянные коэффициенты квадратурной формулы. Формулы типа (5.1) называются формулами Ньютона-Котеса.

Если на интервале (x_0, x_1) функция $f(x)$ интерполируется многочленом первой степени (линейной функцией)

$$L_1(x) = y_0 \frac{x - x_1}{x_0 - x_1} + y_1 \frac{x - x_0}{x_1 - x_0},$$

$$\text{то } A_0 = \int_{x_0}^{x_1} \frac{x - x_1}{x_0 - x_1} dx = \frac{h}{2}, \quad A_1 = \frac{h}{2}, \quad \int_{x_0}^{x_1} f(x) dx = \frac{h(y_0 + y_1)}{2} + R_1.$$

Разбивая интервал (a, b) на n отрезков длиной $h = (b-a)/n$ точками x_0, x_1, \dots, x_n и заменяя функцию $y = f(x)$ на каждом отрезке (x_{i-1}, x_i) , $i = 1, \dots, n$ ее интерполяционным многочленом первой степени, получаем

$$\int_a^b f(x) dx = h(y_0 + y_1)/2 + h(y_1 + y_2)/2 + \dots + h(y_{n-1} + y_n)/2 + R_n$$

$$\text{или} \quad \int_a^b f(x) dx = \frac{(b-a)}{n} \left(y_0/2 + \sum_{i=1}^{n-1} y_i + y_n/2 \right) + R_n. \quad (5.2)$$

Формула (5.2) называется **формулой трапеций**. Ошибка, допускаемая при ее применении, удовлетворяет следующей оценке

$$|R_n| \leq (b-a)h^2 \mu_2 / 12,$$

где μ_2 – максимальное значение $|f''(x)|$ на интервале (a, b) .

Пусть теперь на интервале (x_0, x_1) функция $f(x)$ интерполируется многочленом второй степени

$$\begin{aligned} L_2(x) = & y_0(x-x_1)(x-x_2)/[(x_0-x_1)(x_0-x_2)] + \\ & + y_1(x-x_0)(x-x_2)/[(x_1-x_0)(x_1-x_2)] + \\ & + y_2(x-x_0)(x-x_1)/[(x_2-x_0)(x_2-x_1)]. \end{aligned}$$

В этом случае $x_1 = x_0 + h$; $x_2 = x_0 + 2h$;

$$A_0 = \int_{x_0}^{x_2} \frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} dx = h/3;$$

$$A_1 = 4h/3, \quad A_2 = h/3;$$

$$\int_{x_0}^{x_2} f(x) dx = h(y_0 + 4y_1 + y_2)/3 + R_2.$$

Разбивая интервал (a, b) на $2n$ отрезков длиной $h = (b-a)/2n$ точками x_0, x_1, \dots, x_n и заменяя на каждом отрезке

(x_{2i-2}, x_{2i}) , $i = 1, \dots, n$ функцию интерполяционным многочленом второй степени, получаем

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

$$\text{или} \quad \int_a^b f(x) dx = \frac{(b-a)}{6n}(y_0 + 4\delta_1 + 2\delta_2 + y_{2n}) + R_n, \quad (5.3)$$

$$\text{где } \delta_1 = \sum_{i=1}^n y_{2i-1}, \quad \delta_2 = \sum_{i=1}^{n-1} y_{2i}.$$

Формула (5.3) называется **формулой Симпсона**. Ошибка, допускаемая при ее применении, удовлетворяет следующей оценке:

$$|R_n| < h^4 (b-a) \mu_4 / 180.$$

где μ_4 – максимальное значение $|f^{(4)}(x)|$ на интервале (a, b) .

Схемы алгоритмов вычисления интеграла методом трапеций и методом Симпсона приведены на рис. 5.1.

Пример Вычислить коэффициент n'_σ , учитывающий влияние внешнего трения и сложной формы очага деформации [5, 6], по формуле

$$n'_\sigma = \left\{ \int_{h_*}^{h_0} \left((\delta - 1) \left(\frac{h_0}{h_x} \right)^\delta + 1 \right) dh_x + \int_{h_1}^{h_*} \left((\delta + 1) \left(\frac{h_x}{h_1} \right)^\delta - 1 \right) dh_x \right\} \frac{1}{\Delta h \delta}, \quad (5.4)$$

где h_0 , h_1 – высота полосы до и после клетки (прохода);

h_n – высота полосы в нейтральном сечении (см формулу (3.8));

δ – коэффициент напряженного состояния (см формулу (3.9)).

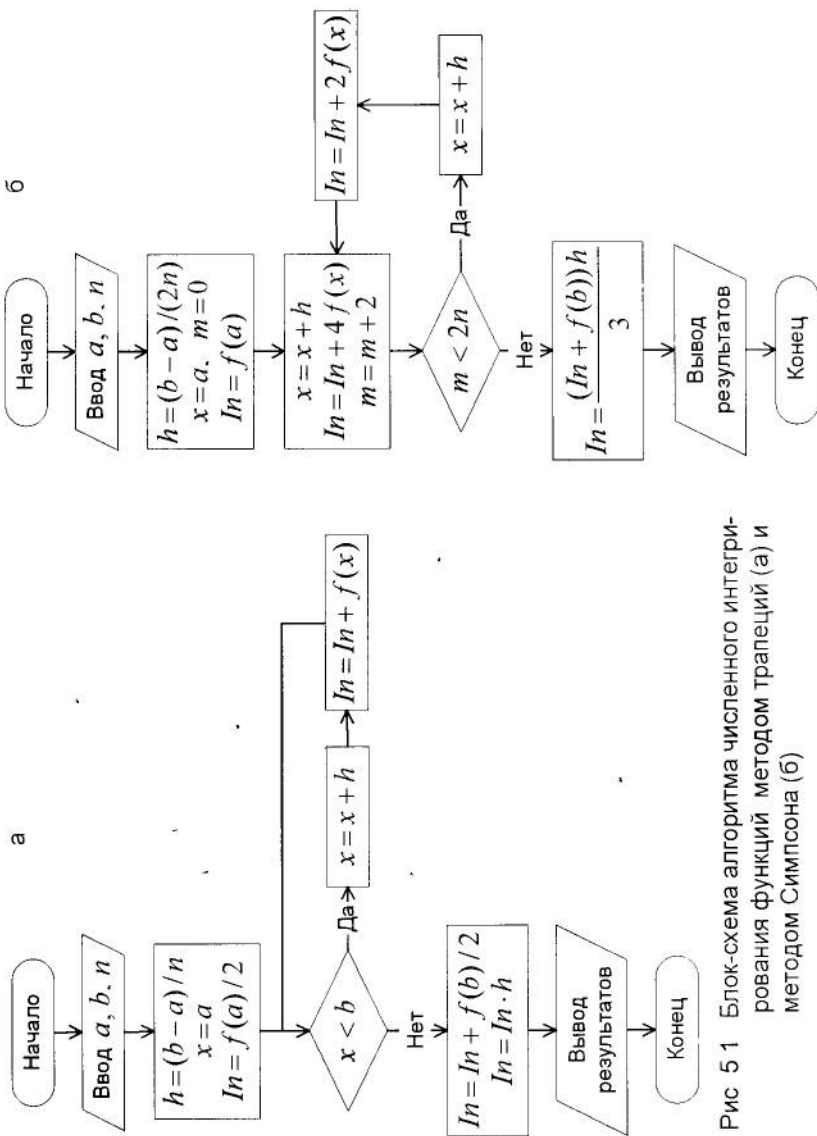


Рис 5.1 Блок-схема алгоритма численного интегрирования функций методом трапеций (а) и методом Симпсона (б)

Лабораторная работа № 10

ЧИСЛЕННОЕ ИНТЕГРИРОВАНИЕ ДИФФЕРЕНЦИАЛЬНЫХ УРАВНЕНИЙ

Причины, по которым прибегают к численным методам решения дифференциальных уравнений, аналогичны причинам, вызывающим численное интегрирование функций: невозможность получить решение уравнения в замкнутом виде или же громоздкость аналитического выражения.

Наиболее простыми методами численного интегрирования обыкновенных дифференциальных уравнений являются методы Эйлера и Рунге-Кутты, которые рассмотрит первоначально применительно к уравнениям первого порядка.

Задача состоит в численном решении дифференциального уравнения $y' = f(x, y)$ с начальным условием $y(x_0) = y_0$, то есть в определении приближенных значений y_1, y_2, \dots, y_n функции в узлах интегрирования x_1, x_2, \dots, x_n .

Предполагая расположение узлов таким, что $x_{i+1} = x_i + h$, $i = 1, \dots, n-1$, где h – шаг интегрирования, согласно формуле Тейлора, получаем

$$\begin{aligned} y_{i+1} &= y(x_{i+1}) = y(x_i) + hy'(x_i) + O(h^2) = \\ &= y(x_i) + hf(x_i, y_i) + O(h^2), \end{aligned}$$

откуда следует приближенное равенство

$$y_{i+1} \approx y_i + hf(x_i, y_i), \quad i = 1, \dots, n-1 \quad (5.5)$$

Метод Эйлера основывается на формуле (5.5) и состоит в том, что задавая значения x_1, x_2, \dots, x_n через промежутки длиной h , вычисляем последовательные значения функции y_1, y_2, \dots, y_n .

При выводе формулы (5.5) была отброшена бесконечно малая порядка h^2 и, следовательно, допускаемая на каждом шаге ошибка пропорциональна h^2 . Достоинством метода Эйлера является простота вычислительного алгоритма, однако, ему присущи малая точность и систематическое накопление ошибок, допущенных на каждом шаге.

В методе Эйлера движение в каждой следующей точке происходит по касательной к кривой, проведенной в предыдущей точке.

Кривизна линии на участке между соседними точками не учитывается, чем и обусловлена низкая точность этого метода. Попытки повысить точность за счет изменения угла наклона траектории вычислений привели к созданию **метода Рунге-Кутты**. Этот метод сохраняет основное достоинство метода Эйлера – его одноступенчатость, когда для вычисления y_{k+1} используется информация лишь в предыдущей точке x_k, y_k , но дает согласование получаемого численного решения с разложением точного решения в ряд Тейлора вплоть до членов порядка h^p , где $p > 2$ (для метода Эйлера $p = 2$).

Наиболее широко при вычислениях на ЭВМ применяется следующая расчетная схема метода Рунге-Кутты:

$$y_{i+1} = y_i + h(k_1 + 2k_2 + 2k_3 + k_4)/6. \quad (5.6)$$

$$\text{где } k_1 = f(x_i, y_i), \quad k_2 = f(x_i + h/2, y_i + k_1 h/2), \\ k_3 = f(x_i + h/2, y_i + k_2 h/2); \quad k_4 = f(x_i + h, y_i + k_3 h)$$

Ошибка, допускаемая при решении дифференциальных уравнений этим методом, пропорциональна по значению h^5 . Вычисление погрешности связано с вычислением производных функции f , что затрудняет ее определение.

На практике для учета погрешностей обычно применяют двойной пересчет: последовательно вычисляют значения y_i с шагом h и с шагом $h/2$. Затем значения $y_{i,h}$ и $y_{i,h/2}$ сравнивают.

Если для заданного $\varepsilon > 0$ $|y_{i,h} - y_{i,h/2}| \leq \varepsilon$, то переходят к расчету y_{i+1} , в противном случае шаг h на i -м шаге уменьшают вдвое и повторяют вычисления $y_{i,h}$ и $y_{i,h/2}$, и так до достижения требуемой точности.

В качестве искомого решения в точке x_i берут $y_i = (y_{i,h} + y_{i,h/2})/2$. Причем выписывают в окончательный результат значения $y_i = y(x_i)$ только в тех точках x_i , которые были получены с первоначальным шагом h , опуская все промежуточные $y(x_i)$, появляющиеся в процессе деления шага h .

Метод Рунге-Кутты обобщается для решения дифференциальных уравнений высших порядков и систем дифференциальных уравнений. Пусть требуется найти численное решение уравнения

при начальных условиях $y(x_0)$, $y'(x_0) = y_1, \dots, y^{(n-1)}(x_0) = y_{n-1}$.

$$\begin{aligned} W_1' &= W_2; \\ W_{n-1}' &= W_n; \\ W_n' &= f(x, W_1, \dots, W_n) \end{aligned}$$

$W_n(x_0) = y_{n-1}$ Таким образом, решение дифференциального уравнения n -го порядка приведено к интегрированию системы n уравнений первого порядка.

$$\begin{aligned} y_1' &= f_1(x, y_1, \dots, y_n); \\ &\vdots \\ y_n' &= f_n(x, y_1, \dots, y_n). \end{aligned} \quad (5.7)$$
$$y_{l,j+1} = y_{l,j} + h(k_{1l} + 2k_{2l} + 2k_{3l} + k_{4l})/6,$$

где $k_{1i} = f_i(x_i, y_i)$, $k_{2i} = f_i(x_i + h/2, y_i + k_{1i}h/2)$;

37

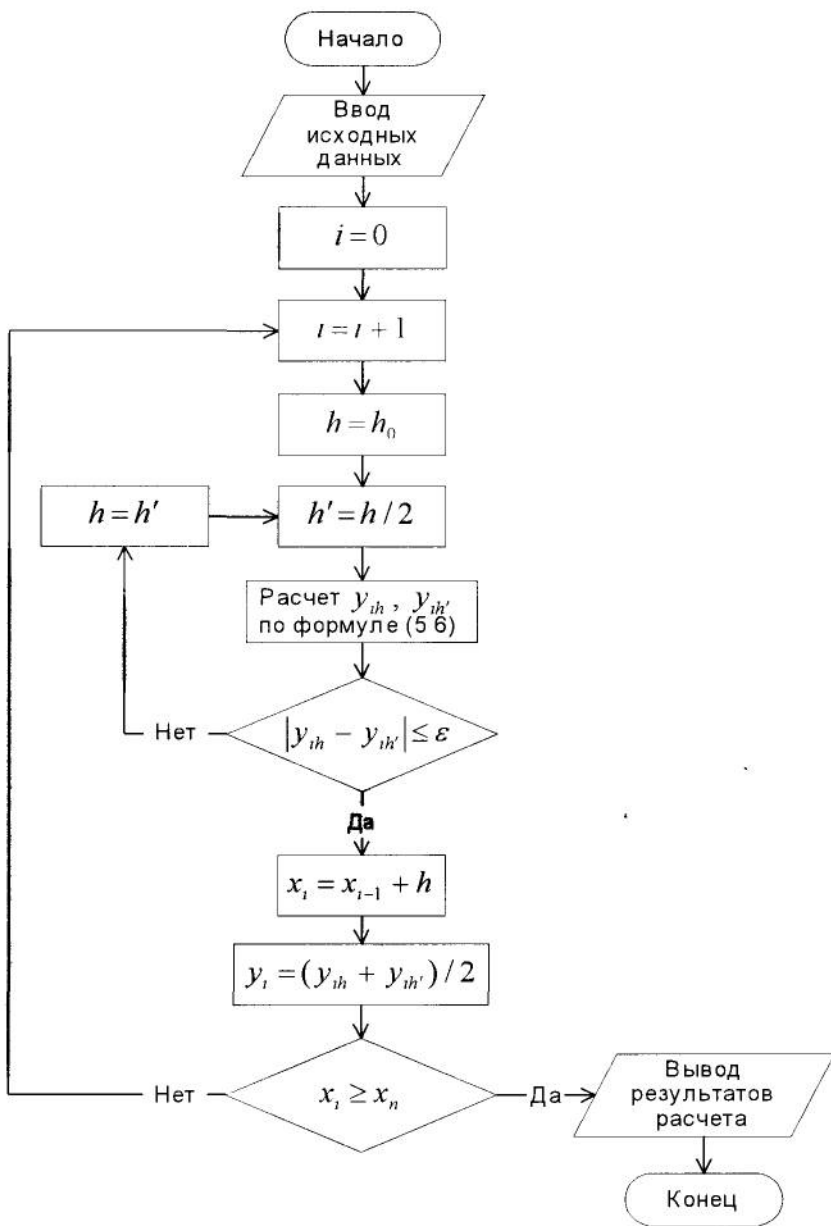


Рис 5.2. Блок-схема алгоритма решения обыкновенных дифференциальных уравнений методом Рунге-Кутты

Пример. Решить уравнение, описывающее изменение нормального давления p_x вдоль очага деформации при тонколистовой холодной прокатке в зависимости от толщины полосы h_x .

$$p'_x = (2K \pm \delta p_x) / h_x, \quad (5.8)$$

где $2K = 1,15\sigma_T$ – вынужденный предел текучести при плоской деформации (прокатка без уширения);

$\delta = \frac{2fl}{\Delta h}$ – коэффициент напряженного состояния;

f – коэффициент контактного трения;

l – длина очага деформации;

$\Delta h = h_0 - h_1$ – абсолютное обжатие полосы;

h_0 и h_1 – толщина полосы до и после клетки.

Знак “плюс” в уравнении (5.8) относится к зоне опережения в очаге деформации, знак “минус” – к зоне отставания. Граничные условия в случае прокатки без натяжения:

1) $p_x = 2K$ при $h_x = h_0$;

2) $p_x = 2K$ при $h_x = h_1$.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Турбо Паскаль 7.0: Учебник. Киев: Торгово-издательское бюро BVH, 1996. 448 с.
2. Дьяконов В.П. Справочник по алгоритмам и программам на языке БЕЙСИК для персональных ЭВМ: Справочник. М.: Наука, 1987. 240 с.
3. Основы применения ЭВМ: Учеб. пособие для вузов / В.П. Каранчук, И.Н. Сваровский, И.Д. Суздальский. М.: Радио и связь, 1988. 288 с.
4. Самохин А.Б., Самохина А.С. Численные методы и программирование на ФОРТРАНЕ для персонального компьютера. – М.: Радио и связь, 1996. 224 с.
5. Вильям Дж. Орвис Visual Basic for Application на примерах: Пер. с англ. – М.: БИНОМ. 512 с.
6. Кузьменко В.Г. VBA 2000: – М. ЗАО „Изд-во БИНОМ“, 2000. 408 с.

7. Аркулис Г Э , Дорогобид В.Г. Теория пластичности: Учеб. пособие для вузов. М.: Металлургия, 1987 352 с.
8. Целиков А.И., Никитин Г.С., Рокотян С.Е. Теория продольной прокатки: Учеб. пособие для вузов. М.. Металлургия, 1980 320 с.
9. Куприн М.И., Куприна М.С. Основы теории прокатки: Учебник для техникумов. 2-е изд., перераб. и доп. М.: Металлургия, 1978 184 с.

Приложение 1

ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ ОТЧЕТА

Отчет по лабораторной работе выполняется в отдельной тетради и должен содержать следующее:

1. Название и цель работы.
2. Теоретическую часть.
3. Блок-схему программы.
4. Пример расчета (в виде распечатки на принтере).
5. Выводы по работе.

Приложение 2

ПРИМЕР ОФОРМЛЕНИЯ ОТЧЕТА

Лабораторная работа № 1

ВЫЧИСЛЕНИЕ ПРОСТЫХ ФУНКЦИЙ

Цель работы:

- научиться определять ряд значений функции, рассчитываемой по конечной формуле или в виде электронной таблицы, или с помощью алгоритмического языка, например, VBA;
- сделать заключение о влиянии исходных данных (факторов) на функцию (отклик).

Требуется определить длину очага деформации при простой продольной прокатке (рис. П2.1)

$$l = \sqrt{R\Delta h},$$

где R – радиус рабочих валков, мм;

$\Delta h = h_0 - h_1$ – абсолютное обжатие, мм;

h_0 и h_1 – высота полосы до и после прокатки, мм.

Необходимо вычислить ряд значений l для нескольких значений радиуса рабочих валков R с определенным шагом и резуль-

таты вывести на печать. Алгоритм решения этой задачи предполагает прямые вычисления по заданному аналитическому выражению (рис. П2.2).

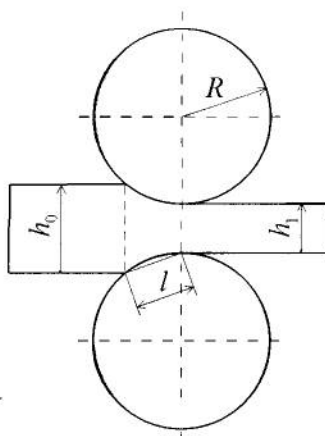


Рис. П2.1. Расчет длины очага деформации при простой продольной прокатке

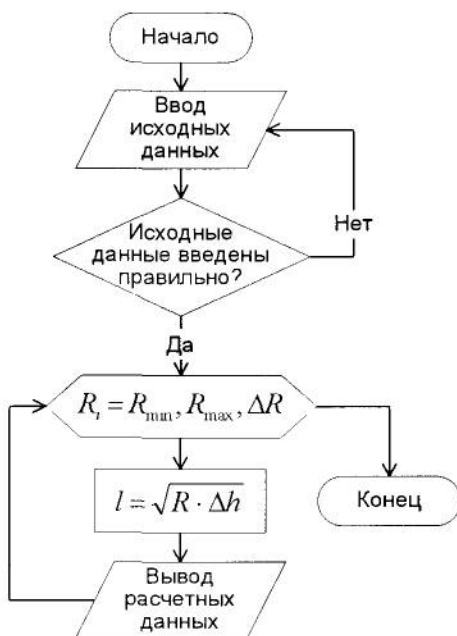


Рис. П2.2. Блок-схема программы расчета нескольких значений функции в цикле

На рис. П2.3 представлен фрагмент рабочего листа **Excel** с результатами расчета длины очага деформации для следующих исходных данных, мм:

$$R_{\min} = 100; R_{\max} = 200; \Delta R = 20;$$

$$h_0 = 10; h_1 = 7, 5.$$

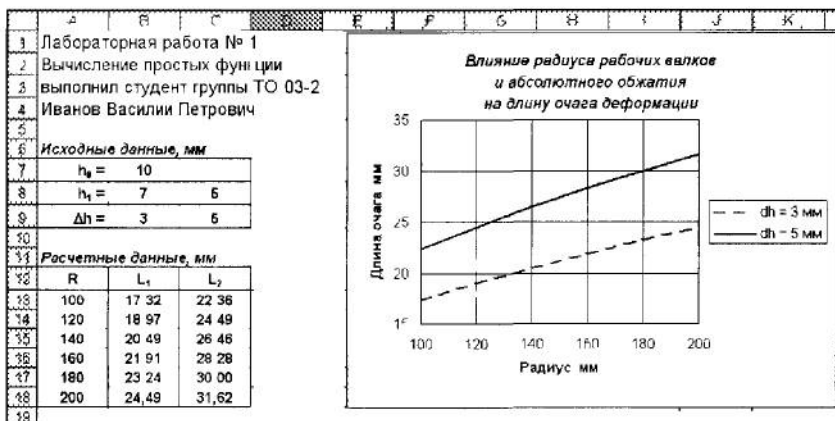


Рис П2 3 Фрагмент рабочего листа с результатами вычислений длины очага деформации

Выводы

- 1 Освоена табличная методика определения ряда значений простых функций в зависимости от исходных параметров
- 2 Длина очага деформации растет с увеличением радиуса рабочих валков и абсолютного обжатия и является нелинейной функцией

Приложение 3

Пример выполнения лабораторной работы № 1 с помощью VBA (рис П3 1 и П3 2)

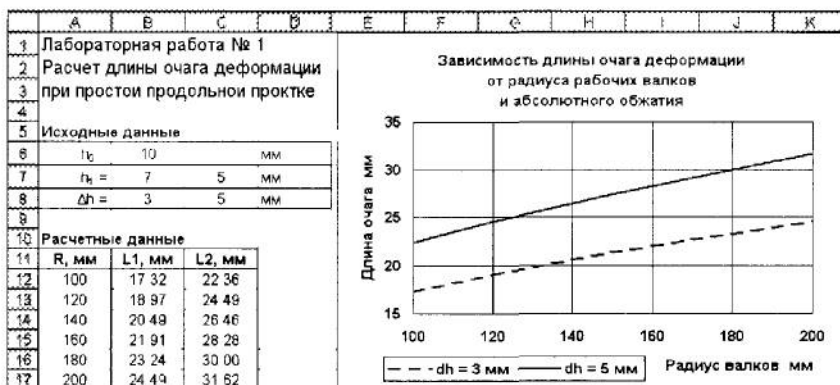


Рис П3 1 Фрагмент рабочего листа с результатами вычислений длины очага деформации

Option Explicit ' Обязательное описание переменных

```
Sub Расчет_длины_очага() ' Название (заголовок) программы (подпрограммы)

Dim h0 As Single, h11 As Single, h12 As Single ' Описание типов переменных
Dim R As Single, Rmin As Single, Rmax As Single, dR As Single ' Описание ...
Dim i As Integer, dh1 As Single, dh2 As Single, L1 As Single, L2 As Single ' Описание ...

Worksheets("VBA").Select ' Выбор нужного листа

h0 = Range("B6").Value ' Ввод данных с листа EXCEL
h11 = Range("B7").Value ' Ввод данных с листа EXCEL
h12 = Range("C7").Value ' Ввод данных с листа EXCEL

Rmin = InputBox("Введите значение Rmin = , мм") ' Ввод из диалогового окна Rmin
Rmax = InputBox("Введите значение Rmax = , мм") ' Ввод из диалогового окна Rmax
dR = InputBox("Введите значение dR = , мм") ' Ввод из диалогового окна dR
' В данной программе ввод исходных данных сделан без проверки

dh1 = h0 - h11: dh2 = h0 - h12 ' Вычисление значений абсолютного обжатия
i = 0 ' Присвоение начального значения вспомогательной переменной

For R = Rmin To Rmax Step dR ' Заголовок цикла
    L1 = (R * dh1) ^ (1 / 2) ' Вычисление 1-го значения длины
    L2 = (R * dh2) ^ (1 / 2) ' Вычисление 2-го значения длины
    Cells(12 + i, 1).Value = R ' Вывод на лист значений R
    Cells(12 + i, 2).Value = L1 ' Вывод на лист значений L1
    Cells(12 + i, 3).Value = L2 ' Вывод на лист значений L2
    i = i + 1 ' Изменение значения вспомогательной переменной
Next R ' Окончание цикла

End Sub ' Конец программы (подпрограммы)
```

Рис. ПЗ.2 Листинг программы VBA «Расчет длины очага деформации» с комментариями

Численные методы Методические указания к лабораторным работам для студентов специальности 150106

Составители *Борис Яковлевич Омельченко*
Михаил Игоревич Румянцев

Редактор Т А Колесникова

Оператор компьютерной верстки Л М Недялкова

Тем план 2006, поз 71

Подписано в печать 31 03 2006

Формат 60×84 1/16

Бумага тип № 1

Плоская печать Усл печ л 3,00

Уч -изд л 3,15

Тираж 100 экз

Заказ 165. Бесплатно



Издательский центр ГОУ ВПО «МГТУ им Г И Носова»

455000, Магнитогорск, пр. Ленина, 38

Полиграфический участок ГОУ ВПО «МГТУ им Г И Носова»