

УДК 519.6

О СТАНДАРТИЗАЦИИ ПРОГРАММ ВЫЧИСЛЕНИЯ ИНТЕГРАЛОВ НА ОСНОВЕ КУСОЧНОЙ ИНТЕРПОЛЯЦИИ ПОДЫНТЕГРАЛЬНЫХ ФУНКЦИЙ

Ромм Я.Е., Джанунц Г.А.

Таганрогский институт имени А.П. Чехова (филиал) ФГБОУ ВО «РГЭУ (РИНХ)», Таганрог,
e-mail: romm@list.ru, janunts@inbox.ru

Компьютерное вычисление определенного интеграла от функции одной вещественной переменной реализуется на основе кусочно-интерполяционного приближения подынтегральной функции. Интерполяция выполняется с помощью полиномов Лагранжа и Ньютона, преобразуемых в каноническую форму алгебраических полиномов с числовыми коэффициентами. Требуемое преобразование осуществляется простым двойным циклом программы. Полученный в результате полином интегрируется, приводя к инвариантным относительно степени полинома формулам Ньютона-Котеса. Коэффициенты формул не зависят от подынтегральной функции, промежутка интегрирования, хранятся в разделе описания констант программы. Пользовательский интерфейс стандартизируется до задания на входе программы подынтегральной функции, промежутка интегрирования, как вариант включает степень полинома и число подынтервалов. Приводится обоснование метода, даны оценки сходимости и скорости сходимости, представлены таблицы коэффициентов для случаев применения полиномов Лагранжа и Ньютона. Описаны коды программ и результаты численного эксперимента, согласно которым на промежутке длины 500 достигается граница абсолютной погрешности вычисления интеграла порядка 10^{-20} . На стандартных промежутках обычной длины достигается нулевая граница погрешности. Метод распространяется на приближение первообразной подынтегральной функции, в этом случае абсолютная погрешность приближения имеет порядок 10^{-19} . Одновременно с минимизацией погрешности минимизируется временная сложность кусочно-интерполяционного вычисления интегралов и первообразных.

Ключевые слова: приближенное вычисление интегралов и первообразных, интерполяционные полиномы Лагранжа и Ньютона, кусочная интерполяция, стандартизация пользовательского интерфейса программ вычисления интегралов

ON STANDARDIZATION OF INTEGRAL CALCULATION PROGRAMS BASED ON PIECEWISE INTERPOLATION OF INTEGRAND FUNCTIONS

Romm Ya.E., Dzhanunts G.A.

Taganrog Branch of the Rostov State University of Economics, Taganrog,
e-mail: romm@list.ru, janunts@inbox.ru

The computer calculation of a certain integral of a function of one real variable is implemented on the basis of a piecewise interpolation approximation of the integrand function. Interpolation is performed with the help of Lagrange and Newton polynomials transformed into the canonical form of algebraic polynomials with numerical coefficients. The required transformation is carried out by a simple double cycle of the program. The resulting polynomial is integrated, leading to Newton-Cotes formulas invariant with respect to the degree of the polynomial. The coefficients of the formulas do not depend on the integrand function, the integration interval, are stored in the description section of the program constants. The user interface is standardized before specifying an integrand function at the input of the program, the integration interval, and as an option includes the degree of the polynomial and the number of subintervals. The justification of the method is given, estimates of convergence and convergence rate are given, tables of coefficients for the cases of Lagrange and Newton polynomials are presented. The program codes and the results of a numerical experiment are described, according to which the limit of the absolute error of calculating the integral of the order of 10^{-20} is reached at the interval of length 500. At standard intervals of normal length, the zero margin of error is reached. The method extends to the approximation of the antiderivative integrand function, in this case the absolute error of the approximation is of 10^{-19} order. Simultaneously with minimizing the error, the time complexity of piecewise interpolation calculation of integrals and antiderivatives is minimized.

Keywords: approximate calculation of integrals and antiderivatives, piecewise interpolation, Lagrange and Newton interpolation polynomials, standardization of the user interface of integral calculation programs

В работе рассматривается компьютерное вычисление определенного интеграла от функции одной вещественной переменной на основе кусочно-интерполяционного приближения подынтегральной функции. Кусочная интерполяция выполняется с помощью интерполяционных полиномов Лагранжа и Ньютона, преобразованных в каноническую форму алгебраических полиномов с числовыми коэффициентами. Апробация подхода была представле-

на в [1], при этом остались нерешенными следующие проблемы. Во-первых, коэффициенты интерполяционных полиномов на каждом подынтервале пересчитывались заново по ходу каждого выполнения программы, что увеличивало трудоемкость метода. В то же время эти коэффициенты можно было записать в память и использовать в разделе констант программы инвариантно относительно вида подынтегральной функции и промежутка интегрирования. Во-

вторых, возникала неточность вычисления коэффициентов полиномов степеней выше пятой вследствие погрешностей операций с плавающей точкой в языке программирования. Это приводило к ограничению набора степеней интерполяционных полиномов, избыточности числа подынтервалов и, как следствие, к ограничениям точности приближения, временной сложности метода, длины промежутков интегрирования. В целом сохранялась общая проблема неустойчивости [2] вычисления интегралов по формулам Ньютона-Котеса высокого порядка, реализуемым на основе кусочной интерполяции с помощью полиномов Лагранжа и Ньютона. Ниже ставится задача устранить отмеченные недостатки подхода к реализации формул Ньютона-Котеса на предложенной основе, в частности требуется повысить точность приближения интеграла, в том числе на больших промежутках интегрирования, и стандартизировать пользовательский интерфейс программ, реализующих метод. Кроме того, ставится задача распространить метод на компьютерное приближение первообразной подынтегральной функции. Требуется выполнить численные эксперименты, описать их ре-

зультаты и привести коды стандартизированных программ.

Цель работы заключается в минимизации погрешности и временной сложности кусочно-интерполяционного вычисления интегралов, а также в стандартизации программ, реализующих метод на основе полиномов Лагранжа и Ньютона. Требуется выполнить математическое обоснование предложенного метода и дать экспериментальное подтверждение его эффективности в рассматриваемых аспектах.

Исходные соотношения. В основе дальнейших интегральных преобразований лежит алгоритм восстановления коэффициентов полинома по его корням. С помощью этого алгоритма преобразуются компоненты полинома, интерполирующего подынтегральную функцию, затем приводятся подобные. В результате интерполяционный полином принимает форму алгебраического полинома с числовыми коэффициентами, что автоматически влечет первообразную от него и упрощает вычисление определенного интеграла. Пусть

$$\sum_{\ell=0}^n d_{\ell} x^{\ell} = \prod_{r=0}^{n-1} (x - x_r). \quad (1)$$

Если $P_1(x) = x - x_0$, то $P_1(x) = d_{11}x + d_{10}$, где $d_{11} = 1$, $d_{10} = -x_0$.

Если уже вычислены коэффициенты полинома

$$P_{k-1}(x) = d_{(k-1)(k-1)}x^{k-1} + d_{(k-1)(k-2)}x^{k-2} + \dots + d_{(k-1)1}x + d_{(k-1)0}, \quad k \geq 2,$$

то $P_k(x) = P_{k-1}(x) \cdot (x - x_{k-1})$.

Приравнивание коэффициентов при одинаковых степенях влечет

$$d_{kk} = d_{(k-1)(k-1)}, \quad d_{k(k-\ell)} = d_{(k-1)(k-\ell-1)} - d_{(k-1)(k-\ell)} x_{k-1}, \quad d_{k0} = -d_{(k-1)0} x_{k-1}, \quad \ell = 1, 2, \dots, k-1. \quad (2)$$

При $k=n$ значения левых частей равенств (2) совпадут с искомыми значениями коэффициентов полинома (1). В матричной форме (2) можно записать в виде

$$\begin{pmatrix} d_{kk} \\ d_{k(k-1)} \\ d_{k(k-2)} \\ \vdots \\ d_{k1} \\ d_{k0} \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ -x_{k-1} & 1 & 0 & \dots & 0 & 0 \\ 0 & -x_{k-1} & 1 & \dots & \vdots & \vdots \\ \dots & \dots & -x_{k-1} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & 1 & 0 \\ \dots & \dots & \dots & \dots & -x_{k-1} & 1 \\ 0 & 0 & 0 & \dots & 0 & -x_{k-1} \end{pmatrix} \times \begin{pmatrix} d_{(k-1)(k-1)} \\ d_{(k-1)(k-2)} \\ \vdots \\ d_{(k-1)1} \\ d_{(k-1)0} \end{pmatrix}.$$

Отсюда следует выражение коэффициентов полинома через его корни [1; 3]

$$\begin{pmatrix} d_n \\ d_{n-1} \\ \dots \\ d_0 \end{pmatrix} = \prod_{\ell=1}^n \begin{pmatrix} 1 & 0 & \dots & 0 & 0 \\ -x_{n-\ell} & 1 & \dots & 0 & 0 \\ 0 & -x_{n-\ell} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & -x_{n-\ell} & 1 \\ 0 & 0 & \dots & 0 & -x_{n-\ell} \end{pmatrix},$$

Представленное восстановление коэффициентов полинома по его корням алгоритмически отличается от формул Виета [4] и не использует уравнения Ньютона для симметрических функций корней [4]. Алгоритм (2) следующим образом применяется к интерполяционному полиному Лагранжа. Для интерполяции функции $y = f(x)$, $x \in [a, b]$ полином Лагранжа можно записать в виде

$$P_n(x) = \sum_{j=0}^n f(x_j) \prod_{\substack{r=0 \\ r \neq j}}^n (x - x_r) / \prod_{\substack{r=0 \\ r \neq j}}^n (x_j - x_r), \tag{3}$$

где x_r – узлы интерполяции. По алгоритму (2) можно вычислить коэффициенты полинома $\prod_{\substack{r=0 \\ r \neq j}}^n (x - x_r)$, затем привести подобные. Непосредственно ниже этот подход применяется для случая равноотстоящих узлов интерполяции.

Пусть на отрезке $x \in [a, b]$ взяты равноотстоящие узлы для полинома (3):

$$x_i \in [a, b], i \in \overline{0, n}, x_{i+1} - x_i = h, i \in \overline{0, n-1}, h = (b - a) n^{-1}.$$

Здесь и всюду в дальнейшем предполагается, что начальный узел совпадает с левой границей отрезка, а конечный узел – с его правой границей. В этом случае $x_j = x_0 + jh$, $x_r = x_0 + rh$, $x_0 = a$, $x_n = b$. Отсюда

$$\prod_{\substack{r=0 \\ r \neq j}}^n (x_j - x_r) = \prod_{\substack{r=0 \\ r \neq j}}^n (j - r)h, P_n(x) = \sum_{j=0}^n f(x_j) \prod_{\substack{r=0 \\ r \neq j}}^n (x - x_r) / (j - r)h^{-1}.$$

Вводится переменная $t = (x - x_0)h^{-1}$.

Тогда $(x - x_1)h^{-1} = t - 1, \dots, (x - x_r)h^{-1} = t - r, r \in \overline{0, n}$, и

$$P_n(t) = \sum_{j=0}^n f(x_j) \prod_{\substack{r=0 \\ r \neq j}}^n (t - r) / (j - r), t = (x - x_0)h^{-1}. \tag{4}$$

В отличие от аналога [5] для (4) ниже выполнен перевод числителей дробей слагаемых в форму полиномов с целочисленными коэффициентами. Как следствие, появится аналитическое выражение первообразной от интерполяционного полинома. Числитель дроби

$\prod_{\substack{r=0 \\ r \neq j}}^n (t - r) / (j - r)$ можно записать в виде

$$P_{nj}(t) = \prod_{r=0}^{n-1} (t - t_r), t_r = \begin{cases} r, & r < j; \\ r + 1, & r \geq j. \end{cases} \tag{5}$$

По схеме (2) получится:

$$P_{nj}(t) = d_{0j} + d_{1j}t + d_{2j}t^2 + \dots + d_{nj}t^n. \tag{6}$$

Коэффициенты полинома (6) – целые числа, они не зависят от интерполируемой функции, от диапазона и расположения её аргумента. Знаменатель дроби в (4) отличается от числителя тем, что в нем $t = j$. В результате

$$P_n(t) = \sum_{j=0}^n f(x_j) (d_{0j} + d_{1j}t + d_{2j}t^2 + \dots + d_{nj}t^n) / (d_{0j} + d_{1j}j + d_{2j}j^2 + \dots + d_{nj}j^n), \quad (7)$$

при $t = (x - x_0)h^{-1}$,

Числитель в (7) вычисляется по схеме Горнера

$$P_{nj}(t) = d_{0j} + d_{1j}t + d_{2j}t^2 + \dots + d_{nj}t^n = (\dots(d_{nj} \times t + d_{n-1j}) \times t + d_{n-2j}) \times t + \dots + d_{1j}) \times t + d_{0j}. \quad (8)$$

В этих обозначениях

$$P_n(t) = \sum_{j=0}^n f(x_j) P_{nj}(t) / P_{nj}(j), \quad t = (x - x_0)h^{-1}. \quad (9)$$

Если каждый коэффициент каждого из полиномов с индексом j в числителях слагаемых (9) разделить на число $P_{nj}(j)$, то интерполяционный полином (7), (9) примет вид

$$P_n(t) = \sum_{j=0}^n f(x_j) (D_{0j} + D_{1j}t + D_{2j}t^2 + \dots + D_{nj}t^n), \quad (10)$$

где

$$D_{lj} = d_{lj} / (d_{0j} + d_{1j}j + d_{2j}j^2 + \dots + d_{nj}j^n), \quad l \in \overline{0, n}. \quad (11)$$

Коэффициенты (11) полинома (10), как и компоненты, из которых они получены, по-прежнему не зависят от вида интерполируемой функции, от отрезка интерполирования, но зависят только от степени интерполяционного полинома и номера узла интерполяции. Этим свойством будут обладать и коэффициенты первообразной полинома (10). Первообразная рассматриваемого полинома примет вид

$$\int P_n(x) dx = h \sum_{j=0}^n f(x_j) (D_{0j}t + 2^{-1}D_{2j}t^2 + \dots + (n+1)^{-1}D_{nj}t^{n+1}) + C, \quad (12)$$

где $t = (x - x_0)h^{-1}$, C – произвольная постоянная.

Приближенное вычисление интегралов на основе кусочной интерполяции с помощью полиномов Лагранжа. Пусть отрезок $[a, b]$ разбит на малые подынтервалы равной длины с общими границами разбиения

$$[a, b] = \bigcup_{i=0}^{p-1} [a_i, b_i], \quad b_i - a_i = (b - a)p^{-1}, \quad a_{i+1} = b_i, \quad i = 0, 1, \dots, p-2. \quad (13)$$

Рассмотренные выше построения воспроизводятся на каждом подынтервале $[a_i, b_i]$ в отдельности, $i \in \overline{0, p-1}$, соответственный полином, его узлы и коэффициенты отмечаются индексом i , для отличия от обозначения других полиномов в обозначение полинома Лагранжа добавляется первая буква фамилии Лагранжа. Полином (10), (11) примет вид

$$P_{Lni}(t) = \sum_{j=0}^n f(x_{ij}) (D_{0j} + D_{1j}t + D_{2j}t^2 + \dots + D_{nj}t^n), \quad [\quad , \quad], \quad (\quad) , \quad (\quad)$$

$$x \in [a_i, b_i], \quad t = (x - a_i)h_i^{-1}, \quad h_i = (b_i - a_i) / n = (b - a) / p / n. \quad (14)$$

Одинаковый на всех подынтервалах шаг интерполяции ниже обозначен

$$h_{pn} = h_i = (b - a) / p / n, \quad \forall i \in \overline{0, p-1}.$$

Непосредственно из (12), (13), (14) вытекают формулы приближенного вычисления интегралов:

$$\int_a^b f(x) dx = \sum_{i=0}^{p-1} \int_{a_i}^{b_i} f(x) dx, \quad \int_{a_i}^{b_i} f(x) dx \approx \int_{a_i}^{b_i} P_{Lni}(x) dx,$$

$P_{Lni}(x) = P_{Lni}(t)$, $P_{Lni}(t)$ определяется из (14).

С введенным обозначением шага интерполяции

$$P_{Lni}(t) = \sum_{j=0}^n f(x_{ij}) (D_{0j} + D_{1j}t + D_{2j}t^2 + \dots + D_{nj}t^n), \quad [a_i, b_i], \quad (i)_{pn}, \quad pn \quad (15)$$

$$x \in [a_i, b_i], \quad t = (x - a_i)h_{pn}^{-1}, \quad h_{pn} = (b - a) / p / n.$$

Из (12) с заменой переменной $t = (x - a_i)h_{pn}^{-1}$ и из (15) следует

$$\int_{a_i}^{b_i} P_{Lni}(x) dx = \sum_{j=0}^n f(x_{ij}) h_{pn} \int_0^n (D_{0j} + D_{1j}t + D_{2j}t^2 + \dots + D_{nj}t^n) dt.$$

Взятие первообразной влечет

$$\int_{a_i}^{b_i} P_{Lni}(x) dx = h_{pn} \sum_{j=0}^n f(x_{ij}) \left(D_{0j} t \Big|_0^n + D_{1j} \frac{1}{2} t^2 \Big|_0^n + D_{2j} \frac{1}{3} t^3 \Big|_0^n + \dots + D_{nj} \frac{1}{n+1} t^{n+1} \Big|_0^n \right),$$

или, в результате выполнения подстановки,

$$\int_{a_i}^{b_i} P_{Lni}(x) dx = h_{pn} \sum_{j=0}^n f(x_{ij}) (D_{0j}n + 2^{-1}D_{1j}n^2 + 3^{-1}D_{2j}n^3 + \dots + (n+1)^{-1}D_{nj}n^{n+1}). \quad (16)$$

Сложение (16) по всем подынтервалам дает формулу приближенного вычисления интеграла:

$$\int_a^b f(x) dx \approx h_{pn} \sum_{i=0}^{p-1} \sum_{j=0}^n f(x_{ij}) (D_{0j}n + 2^{-1}D_{1j}n^2 + 3^{-1}D_{2j}n^3 + \dots + (n+1)^{-1}D_{nj}n^{n+1}), \quad (17) \quad (i)_{pn} / p / n.$$

при $h_{pn} = (b - a) / p / n$.

Как отмечалось, выражение в скобках внутренней суммы в (17) является числом, зависящим от степени интерполяционного полинома и от номера узла интерполяции, но не зависящим от номера подынтервала и вида подынтегральной функции. Ниже это число обозначается

$$c_{nj} = D_{0j}n + 2^{-1}D_{1j}n^2 + 3^{-1}D_{2j}n^3 + \dots + (n+1)^{-1}D_{nj}n^{n+1}. \quad (18)$$

Коэффициенты $D_{\ell j}$, $\ell = 0, 1, \dots, n$, получаются переходом от (5) к (6) в соотношении (4) с использованием (2). Значения (18) могут быть вычислены априори, записаны в память компьютера, быть хранимыми константами и применяться для приближенного вычисления интеграла от любой функции. В частности, их можно сохранять в разделе описаний констант программы и непосредственно использовать в разделе инструкций этой программы при вычислении интегралов вида (17).

Из изложенного вытекает

Теорема 1. На основе кусочной интерполяции подынтегральной функции с помощью полиномов Лагранжа значение интеграла приближенно вычисляется из соотношения

$$\int_a^b f(x) dx \approx (b - a) p^{-1} n^{-1} \sum_{i=0}^{p-1} \sum_{j=0}^n c_{nj} f(x_{ij}), \quad (19)$$

где $f(x_{ij})$ – значение подынтегральной функции в j -м узле интерполяции на i -м подынтервале из (13), коэффициенты и числовые параметры определяются из (17), (18).

Ниже приводится таблица значений коэффициентов (18) (поделенных на n) для различных степеней интерполяционных полиномов Лагранжа.

Замечание 1. В Delphi значения коэффициентов, начиная с полиномов степени $n = 5$, в формате extended вычисляются с погрешностью, этим объясняются ограничения степеней полиномов Лагранжа при вычислении интегралов на основе кусочной интерполяции по программам, представленным в [1]. Для повышения точности вычислений при составлении табл. 1 использовался язык Python с модулем decimal.

Таблица 1

Значения коэффициентов c_{nj} / n приближения интеграла (19) для различных степеней n полинома Лагранжа

n	$c_{nj} / n, j \in \overline{0, n}$
1	(0.50000000000000000000, 0.50000000000000000000)
2	(0.16666666666666666667, 0.66666666666666666667, 0.16666666666666666667)
3	(0.12500000000000000000, 0.37500000000000000000, 0.37500000000000000000, 0.12500000000000000000)
4	(0.07777777777777777778, 0.35555555555555555556, 0.13333333333333333333, 0.35555555555555555556, 0.07777777777777777778)
5	(0.06597222222222222222, 0.26041666666666666667, 0.17361111111111111111, 0.17361111111111111111, 0.26041666666666666667, 0.06597222222222222222)
6	(0.048809523809523809524, 0.257142857142857142857, 0.032142857142857142857, 0.323809523809523809524, 0.032142857142857142857, 0.257142857142857142857, 0.048809523809523809524)
7	(0.043460648148148148148, 0.207002314814814814815, 0.07656250000000000000, 0.172974537037037037037, 0.172974537037037037037, 0.07656250000000000000, 0.207002314814814814815, 0.043460648148148148148)
8	(0.034885361552028218695, 0.207689594356261022928, -0.032733686067019400353, 0.370229276895943562610, -0.160141093474426807760, 0.370229276895943562610, -0.032733686067019400353, 0.207689594356261022928, 0.034885361552028218695)
9	(0.031886160714285714286, 0.175680803571428571429, 0.012053571428571428571, 0.215892857142857142857, 0.064486607142857142857, 0.064486607142857142857, 0.215892857142857142857, 0.012053571428571428571, 0.175680803571428571429, 0.031886160714285714286)
10	(0.026834148361926139704, 0.177535941424830313719, -0.081043570626903960237, 0.454946288279621612955, -0.435155122655122655123, 0.713764630431297097963, -0.435155122655122655123, 0.454946288279621612955, -0.081043570626903960237, 0.177535941424830313719, 0.026834148361926139704)

Равенство коэффициентов c_{nj} и $c_{n(n-j)}$, наблюдаемое в табл. 1, является общим свойством приближения (19), которое коррелируется с известной симметрией коэффициентов формул Ньютона-Котеса [2; 6], выполненных на основе полиномов Лагранжа.

Замечание 2. В компьютерной реализации вычисления интеграла из (19) подынтервалы (13) в суммарном объединении могут не поместиться в точности в заданный промежуток интегрирования $[a, b]$: $b - a$ не обязательно делится нацело на $b_i - a_i$. Целесообразно сначала вычислить интеграл на $[a, b_{p-2}]$ ($b_{p-2} - a = a_{p-1} - a$ нацело делится на $b_i - a_i$), исключив последний подынтервал. Затем на последнем (исключенном) подынтервале $[a_{p-1}, b]$ с реальным значением b отдельно провести вычисление интеграла из (16), заменив h_{pn} на соответствующий шаг интерполяции $h_{pn} = (b - a_{p-1}) / n$. Получится

$$\int_{a_{p-1}}^b P_{Ln(p-1)}(x) dx = \tilde{h}_{pn} \sum_{j=0}^n c_{nj} f(x_{(p-1)j}), \quad (20)$$

c_{nj} по-прежнему из (18). Значение (20) складывается с интегралом по промежутку $[a, b_{p-2}]$, что дает окончательное приближение интеграла на $[a, b]$.

При данной коррекции на последнем подынтервале начальные и конечные узлы интерполяции в точности совпадут с его границами, и удастся избежать часто возникающей погрешности компьютерной реализации квадратурных формул [1].

Программная реализация кусочно-интерполяционного вычисления интеграла на основе полиномов Лагранжа, соотношений (17) – (19) и коэффициентов из табл. 1 приводится непосредственно ниже (здесь и в дальнейшем Delphi 9 или Delphi 10).

```

program PIL_integral_n1_10; {$APPTYPE CONSOLE} uses SysUtils; var a,b,S:extended; n:byte; p:int64;
function f(x:extended):extended;
begin f:=cos(x)*exp(sin(x)){cos(x)}{sqrt(1-0.5*sqr(sin(x)))}{exp(x/2)+cos(4*x)}{x*exp(-x)*cos(2*x)} end;
function exact_int(a,b:extended):extended;
begin exact_int:=exp(sin(b))-exp(sin(a)){sin(b)-sin(a)}{1.35064388104767550252}{2*(exp(Pi)-1)}
{1/25*(exp(-2*Pi)*(3-10*Pi)-3)} end;
function PIL_int(a,b:extended; n:byte; p:int64):extended;
const c:array[1..10, 0..5] of extended =((0.5, 0, 0, 0, 0, 0), (0.16666666666666666667,
0.66666666666666666667, 0, 0, 0, 0), (0.125, 0.375, 0, 0, 0, 0), (0.07777777777777777778,
0.35555555555555555556, 0.13333333333333333333, 0, 0, 0), (0.06597222222222222222,
0.26041666666666666667, 0.17361111111111111111, 0, 0, 0),
(0.048809523809523809524, 0.257142857142857142857, 0.032142857142857142857,
0.323809523809523809524, 0, 0), (0.043460648148148148148, 0.207002314814814815, 0.0765625,
0.172974537037037037037, 0, 0), (0.034885361552028218695, 0.207689594356261022928,
-0.032733686067019400353, 0.370229276895943562610,
-0.160141093474426807760, 0), (0.031886160714285714286, 0.175680803571428571429,
0.012053571428571428571, 0.215892857142857142857, 0.064486607142857142857, 0),
(0.026834148361926139704, 0.177535941424830313719, -0.081043570626903960237,
0.454946288279621612955, -0.435155122655122655123, 0.713764630431297097963));
type coeffs = array[0..5] of extended; x_arr = array[0..10] of extended;
var ai,hp,hpn,s1,s2:extended; j:byte;i:int64; s2_arr:coeffs; x:x_arr;
begin hp:=(b-a)/p; hpn:=hp/n; s1:=0; ai:=a; i:=0; for j:=1 to n div 2 do s2_arr[j]:=0;
repeat
for j:=1 to n do x[j]:=ai+j*hp; for j:=1 to n div 2 - 1 do s2_arr[j]:=s2_arr[j]+f(x[j])+f(x[n-j]);
if n mod 2 <>0 then s2_arr[n div 2]:=s2_arr[n div 2]+f(x[n div 2])+f(x[n div 2 + 1])
else s2_arr[n div 2]:=s2_arr[n div 2]+f(x[n div 2]); if i < p-1 then s1:=s1+f(x[n]);
inc(i); ai:=a+i*hp;
until i = p;
s2:=0; for j:=1 to n div 2 do s2:=s2+c[n,j]*s2_arr[j]; s1:=c[n,0]*(f(a)+2*s1+f(b)); PIL_int:=(s1+s2)*hp;
end;
begin
a:=0; b:=Pi/2; n:=10; p:=16; S:=PIL_int(a, b, n, p); writeln('S=',S,' Pogr=',abs(S-exact_int(a,b))); readln;
end.
    
```

Результаты численного эксперимента на основе работы программы PIL_integral_n1_10 приведены в табл. 2.

Таблица 2

Значения параметров и погрешности вычисления интегралов по результатам численного эксперимента

Интеграл	Приближение интеграла	Абсолютная погрешность приближения	Параметры программы
$\int_0^{\pi/2} \cos(x) e^{\sin(x)} dx$	1.71828182845905E+0000	0.00000000000000E+0000	$n = 5, p = 512$
$\int_0^{500} \cos(x) e^{\sin(x)} dx$	-3.73603552314934E-0001	5.42101086242752E-0020	$n = 9, p = 4096$
$\int_0^{\pi/2} \cos(x) dx$	1.00000000000000E+0000	0.00000000000000E+0000	$n = 6, p = 32$
$\int_0^{\pi/2} \sqrt{1-2^{-1} \sin^2(x)} dx$	1.35064388104768E+0000	0.00000000000000E+0000	$n = 2, p = 64$
$\int_0^{2\pi} (e^{x/2} + \cos(4x)) dx$	4.42813852655585E+0001	0.00000000000000E+0000	$n = 5, p = 1024$
$\int_0^{2\pi} (x e^{-x} \cos(2x)) dx$	-1.22122604618968E-0001	0.00000000000000E+0000	$n = 7, p = 4096$

Во втором столбце табл. 1 – вычисленное программой PIL_integral_n1_10 приближенное значение интеграла из первого столбца той же строки, в третьем столбце – абсолютная погрешность приближения, в четвертом – параметры программы. Для полного эллиптического интеграла второго рода $E(1/\sqrt{2}) = \int_0^{\pi/2} \sqrt{1-2^{-1}\sin^2(x)} dx$ в качестве эталонного

значения для сравнения принято значение $E(1/\sqrt{2})$, представленное в [7].

Вычисление интеграла из (19), включая возможное уточнение (20), обладает естественным параллелизмом. Для одновременного выполнения всех умножений вида $c_{nj}f(x_{ij})$ за время t_y одного бинарного умножения потребуется $R = p(n+1)$ процессорных элементов (ПЭ). Сумма парных произведений вычисляется по схеме сдваивания [8], что потребует $\lfloor R/2 \rfloor$ ПЭ и займет время $\lfloor \log_2 p + \log_2(n+1) \rfloor t_c$, t_c – время одного бинарного сложения. Здесь и ниже $\lfloor \beta \rfloor$ обозначает число, ближайшее целое к β не меньшее β . Требуется еще умножение полученной суммы на h_{pn} . Вычисление шага $h_{pn} = (b-a)p^{-1}n^{-1}$ можно выполнить по ходу схемы сдваивания. В результате временная сложность $T(R)$ параллельного вычисления интеграла составит

$$\begin{aligned} T(p(n+1)) &= 2t_y + \lfloor \log_2 p + \log_2(n+1) \rfloor t_c \sim (\log_2 p + \log_2(n+1)) t_c = \\ &= O((\log_2 p + \log_2(n+1))). \end{aligned} \quad (21)$$

При ограничении степени полинома, $n \leq n_0 = \text{const}$, –

$$T(p(n+1)) = O(\log_2 p).$$

Вычисление интеграла на основе первообразной от непрерывного кусочно-интерполяционного приближения подынтегральной функции. Пусть на отрезке (13) выполнено приближение (15) подынтегральной функции,

$$f(x) \approx P_{Lni}(t), \quad x \in [a, b] \cap [a_i, b_i].$$

Номер подынтервала, при котором $x \in [a_i, b_i]$, $t = (x-a_i)h_{pn}^{-1}$, h_{pn} из (17), дает формула [1]

$$i = \lfloor (x-a) / ((b-a)p^{-1}) \rfloor,$$

$\lfloor a \rfloor$ – целая часть числа a . Пусть функция $f(x)$ непрерывна $\forall x \in [a, b]$ из (13), тогда она интегрируема, ее первообразная

$$\Phi(x) = \int_a^x f(x) dx \quad (22)$$

непрерывна на $[a, b]$ [9]. Приближение подынтегральной функции, обозначаемое $F_{Ln}(x)$, рассматривается как самостоятельная функция, определенная на всем $[a, b]$,

$$F_{Ln}(x) \approx f(x) \quad \forall x \in [a, b],$$

при этом определяющими являются соотношения:

$$\forall x \in [a, b], \forall i = 0, 1, \dots, p-1: \quad x \in [a_i, b_i] \Rightarrow F_{Ln}(x) = P_{Lni}(t), \quad t = (x-a_i)h_{pn}^{-1}, \quad (23)$$

$P_{Lni}(t)$ из (15).

Начальный и конечный узел интерполяции полинома (15) всегда находятся соответственно на левой и правой границе подынтервала, поэтому при $i = 0, 1, \dots, p-2$

$$F_{Ln}(a_{i+1}) = P_{Ln(i+1)}(t) \Big|_{x=a_{i+1}} = P_{Ln(i+1)}(0) = f(a_{i+1}) = f(b_i) = P_{Lni}(n) = P_{Lni}(t) \Big|_{x=b_i} = F_{Ln}(b_i). \quad (24)$$

Согласно (15), (23), (24) функция $F_{Ln}(x)$ непрерывна $\forall x \in [a, b]$, следовательно, интегрируема на $[a, b]$. Ее первообразная

$$\Phi_{Ln}(x) = \int_a^x F_{Ln}(x) dx + C \quad (25)$$

при $\forall C = \text{const}$ определена и непрерывна $\forall x \in [a, b]$.

Первообразную (25) можно построить из полиномов (15).

В случае $x \in [a_i, b_i]$ интеграл

$$\int_{a_i}^x P_{Lni}(t) dx = \sum_{j=0}^n f(x_{ij}) h_{pn} \int_{a_i}^x (D_{0j} + D_{1j}t + D_{2j}t^2 + \dots + D_{nj}t^n) dt. \quad (26)$$

представляет собой первообразную полинома (15). Равенство (26) эквивалентно равенству

$$\int_{a_i}^x P_{Lni}(t) dx = \sum_{j=0}^n f(x_{ij}) h_{pn} \left(D_{0j} t \Big|_{a_i}^x + 2^{-1} D_{1j} t^2 \Big|_{a_i}^x + 3^{-1} D_{2j} t^3 \Big|_{a_i}^x + \dots + (n+1)^{-1} D_{nj} t^{n+1} \Big|_{a_i}^x \right). \quad (27)$$

Для полинома (27) вводится обозначение при $x \in [a_i, b_i]$

$$P_{INTLni}(x) = \sum_{j=0}^n f(x_{ij}) h_{pn} \left(D_{0j} t \Big|_{a_i}^x + 2^{-1} D_{1j} t^2 \Big|_{a_i}^x + 3^{-1} D_{2j} t^3 \Big|_{a_i}^x + \dots + (n+1)^{-1} D_{nj} t^{n+1} \Big|_{a_i}^x \right), \quad (28)$$

так что

$$\int_{a_i}^x P_{Lni}(t) dx = P_{INTLni}(x). \quad (29)$$

В (28) каждая нижняя подстановка равна нулю, поэтому

$$\int_{a_r}^{b_r} P_{Lnr}(t) dx = \int_{a_r}^{a_{r+1}} P_{Lnr}(t) dx = P_{INTLnr}(a_{r+1}) \quad \forall r = 0, 1, \dots, p-1. \quad (30)$$

Из (29), (30) следует

$$\int_{a_i}^x P_{Lni}(t) dx + \int_{a_{i-1}}^{a_i} P_{Ln(i-1)}(t) dx = P_{INTLni}(x) + P_{INTLn(i-1)}(a_i). \quad (31)$$

С учетом (23)

$$\int_{a_i}^x F_{Ln}(x) dx + \int_{a_{i-1}}^{a_i} F_{Ln}(x) dx = P_{INTLni}(x) + P_{INTLn(i-1)}(a_i),$$

или, с учетом (13),

$$\int_{a_{i-1}}^x F_{Ln}(x) dx = P_{INTLni}(x) + P_{INTLn(i-1)}(b_{i-1}). \quad (32)$$

Аналогично, добавление к обеим частям (31) интеграла $\int_{a_{i-2}}^{a_{i-1}} P_{Ln(i-2)}(t) dx$ влечет

$$\int_{a_i}^x P_{Lni}(t) dx + \int_{a_{i-1}}^{a_i} P_{Ln(i-1)}(t) dx + \int_{a_{i-2}}^{a_{i-1}} P_{Ln(i-2)}(t) dx = P_{INTLni}(x) + P_{INTLn(i-1)}(a_i) + P_{INTLn(i-2)}(a_{i-1}).$$

С учетом определения (23) функции $F_{Ln}(x)$,

$$\int_{a_i}^x F_{Ln}(x) dx + \int_{a_{i-1}}^{a_i} F_{Ln}(x) dx + \int_{a_{i-2}}^{a_{i-1}} F_{Ln}(x) dx = P_{INTLni}(x) + P_{INTLn(i-1)}(a_i) + P_{INTLn(i-2)}(a_{i-1}),$$

или, аналогично (32),

$$\int_{a_{i-2}}^x F_{Ln}(x) dx = P_{INTLni}(x) + P_{INTLn(i-1)}(b_{i-1}) + P_{INTLn(i-2)}(b_{i-2}).$$

Процесс можно продолжать до получения индекса $i - i = 0$ в нижнем пределе интегрирования. Поскольку $a_0 = a$, в продолжение процесса получится при $x \in [a_i, b_i]$

$$\int_a^x F_{Ln}(x) dx = P_{INTLn_i}(x) + P_{INTLn(i-1)}(b_{i-1}) + P_{INTLn(i-2)}(b_{i-2}) + P_{INTLn(i-3)}(b_{i-3}) + \dots + P_{INTLn_0}(b_0). \quad (33)$$

Из изложенного вытекает

Лемма 1. Первообразная (22) может вычисляться с помощью непрерывного кусочно-интерполяционного приближения (23) подынтегральной функции из соотношения

$$\Phi(x) \approx \int_a^x F_{Ln}(x) dx = P_{INTLn_i}(x) + \sum_{\ell=1}^i P_{INTLn(i-\ell)}(b_{i-\ell}), \quad x \in [a_i, b_i], \quad (34)$$

где $P_{INTLn_i}(x)$ из (28), $[a_i, b_i]$ – из (13), $P_{INTLn_r}(b_r)$ вычисляется из (28) при $i = r, x = b_r$.

Первообразная (25), (33) представляет собой непрерывное аналитическое приближение первообразной подынтегральной функции (22), если первообразная от последней аналитического выражения не имеет. Из (33), (34) при $x = b$ получается формула приближенного вычисления интеграла

$$\int_a^b F_{Ln}(x) dx = P_{INTLn(p-1)}(b_{p-1}) + P_{INTLn(p-2)}(b_{p-2}) + \dots + P_{INTLn(i-1)}(b_{i-1}) + P_{INTLn(i-2)}(b_{i-2}) + \dots + P_{INTLn_1}(b_1) + P_{INTLn_0}(b_0). \quad (35)$$

Более точно, имеет место

Теорема 2. В условиях леммы 1

$$\int_a^b f(x) dx \approx \int_a^b F_{Ln}(x) dx = \sum_{i=0}^{p-1} P_{INTLn_i}(b_i), \quad (36)$$

где $F_{Ln}(x)$ определяется из (23), $P_{INTLn_i}(b_i)$ вычисляется из (28) при $x = b_i$.

В обозначении (25) первообразная (33) примет вид

$$\Phi_{Ln}(x) = P_{INTLn_i}(x) + \sum_{\ell=1}^i P_{INTLn(i-\ell)}(b_{i-\ell}), \quad \forall x \in [a_i, b_i], \quad \forall i \in \overline{0, p-1}. \quad (37)$$

Для (37) по основной теореме интегрального исчисления [9] должно выполняться равенство

$$\int_a^b F_{Ln}(x) dx = \Phi_{Ln}(b) - \Phi_{Ln}(a). \quad (38)$$

Более подробно,

$$\begin{aligned} \int_a^b F_{Ln}(x) dx &= P_{INTLn(p-1)}(b_{p-1}) + P_{INTLn(p-2)}(b_{p-2}) + \dots + P_{INTLn(i-1)}(b_{i-1}) + \\ &+ P_{INTLn(i-2)}(b_{i-2}) + \dots + P_{INTLn_1}(b_1) + P_{INTLn_0}(b_0) - P_{INTLn(p-1)}(a_{p-1}) - \\ &- P_{INTLn(p-2)}(a_{p-2}) - \dots - P_{INTLn(i-1)}(a_{i-1}) - P_{INTLn(i-2)}(a_{i-2}) - \dots - P_{INTLn_1}(a_1) - P_{INTLn_0}(a_0). \end{aligned} \quad (39)$$

Из (28) $P_{INTLn_r}(a_r) = 0, r = 0, 1, \dots, p-1$, и из (38), (39)

$$\Phi_{Ln}(b) - \Phi_{Ln}(a) = \Phi_{Ln}(b) = \int_a^b F_{Ln}(x) dx = \sum_{i=0}^{p-1} P_{INTLn_i}(b_i).$$

Окончательно,

$$\int_a^b f(x) dx \approx \Phi_{Ln}(b) = \int_a^b F_{Ln}(x) dx = \sum_{i=0}^{p-1} P_{INTLn_i}(b_i),$$

что является разновидностью записи (36).

Для коэффициентов полинома $P_{INTLn_i}(x)$ из (28) вводится обозначение

$$D_{INT\ell_j} = (\ell + 1)^{-1} D_{\ell_j}, \ell = 0, 1, \dots, n. \tag{40}$$

Полином (28) примет вид

$$P_{INTLn_i}(x) = h_{pn} \sum_{j=0}^n f(x_{i_j}) \sum_{\ell=0}^n D_{INT\ell_j} t^{\ell+1} \Big|_{a_i}^x, \quad x \in [a_i, b_i], i \in \overline{0, p-1}. \tag{41}$$

В качестве примера в табл. 3 даны коэффициенты (40) полинома (41), поделенные на n , для степени $n = 4$.

Таблица 3

Значения коэффициентов $D_{INT\ell_j} / n$ для приближения первообразной в случае полинома Лагранжа степени $n = 4$

i	$D_{INT\ell_j} / n, \ell \in \overline{0, n}$
0	(1, -1.041666666666666666666666666666667E+0, 4.8611111111111111111111111111111E-1, -1.041666666666666666666666666666667E-1, 8.3333333333333333333333333333333E-3)
1	(0, 2.0000000000000000000000000000000E+0, -1.4444444444444444444444444444444E+0, 3.7500000000000000000000000000000E-1, -3.3333333333333333333333333333333E-2)
2	(0, -1.5000000000000000000000000000000E+0, 1.5833333333333333333333333333333E+0, -5.0000000000000000000000000000000E-1, 5.0000000000000000000000000000000E-2)
3	(0, 6.6666666666666666666666666666666E-1, -7.77777777777777777777777777777E-1, 2.9166666666666666666666666666666E-1, -3.3333333333333333333333333333333E-2)
4	(0, -1.2500000000000000000000000000000E-1, 1.52777777777777777777777777777E-1, -6.2500000000000000000000000000000E-2, 8.3333333333333333333333333333333E-3)

При вычислении (41) используется схема (8) для внутренней суммы, затем находится внешняя сумма парных произведений. Вычисление (37) сводится к сложению значения (41) с суммой значений аналогичных полиномов на правых границах предшествующих подынтервалов. В последовательном варианте можно переходить от подынтервала к подынтервалу, добавляя к текущему значению $P_{INTLn_i}(b_i)$ суммы по одному слагаемому $P_{INTLn(i-1)}(b_{i-1})$ с границы предшествующего подынтервала. При $x = b$ определяется приближение интеграла.

Замечание 3. Для рассматриваемого варианта метода можно повторить замечание 2: на последнем подынтервале следует заново рассчитать шаг интерполяции $\tilde{h}_{pn} = (b - a_{p-1}) / n$, взяв точные границы подынтервала.

Программа вычисления интеграла на основе первообразной от кусочно-интерполяционного приближения подынтегральной функции с помощью полинома Лагранжа и коэффициентов табл. 3 представлена непосредственно ниже.

```

program PIL_integral_n4_perv; {$APPTYPE CONSOLE} uses SysUtils; var a,b,S:extended; p:int64;
function f(x:extended):extended;
begin f:=cos(x)*exp(sin(x)){cos(x)} {sqrt(1-0.5*sqr(sin(x)))} {exp(x/2)+cos(4*x)} {x*exp(-x)*cos(2*x)} end;
function exact_int(a,b:extended):extended;
begin exact_int:=exp(sin(b))-exp(sin(a)){sin(b)-sin(a)} {1.35064388104767550252} {2*(exp(Pi)-1)}
{1/25*(exp(-2*Pi)*(3-10*Pi)-3)} end;
function PIL_int(a,b:extended; p:int64):extended;
const n=4; Dint:array[0..n,0..n] of extended =
((1, -1.041666666666666666666666666666667, 4.8611111111111111111111111111111E-1, -1.041666666666666666666666666666667E-1,
8.3333333333333333333333333333333E-3), (0, 2, -1.4444444444444444444444444444444E+0, 3.75E-1,
-3.3333333333333333333333333333333E-2),
(0, -1.5, 1.5833333333333333333333333333333, -5.0E-1, 5.0E-2), (0, 6.6666666666666666666666666666666E-1,
-7.77777777777777777777777777777E-1, 2.9166666666666666666666666666666E-1, -3.3333333333333333333333333333333E-2),
(0, -1.25E-1, 1.52777777777777777777777777777E-1, -6.25E-2, 8.3333333333333333333333333333333E-3));
var ai,hp,hpn,s,pp2:extended; j,r:byte;i:int64; c:array[0..n] of extended;
begin for j:=0 to n do begin pp2:= Dint[j][n]; for r:=1 to n do pp2 := pp2*n + Dint[j][n-r]; c[j]:=pp2; end;
hp:=(b-a)/p; hpn:=hp/n; i:=0; ai:=a; s:=0;
repeat for j:=0 to n do s:=s+c[j]*(f(ai+j*hp)); inc(i); ai:=a+i*hp; until i = p;
PIL_int:=s*hp; end;
begin
a:=0; b:=Pi/2; p:=1024; S:=PIL_int(a, b, p); writeln('S=', S, ' Pogr=', abs(S-exact_int(a,b))); readln;
end.
    
```

Реализация дана для $n = 4$, повышение степени интерполяционного полинома к улучшению точности не приводит. Результаты численных экспериментов с программой PII_integral_n4_perv приведены в табл. 4.

Таблица 4

Результаты приближенного вычисления интегралов
с помощью программы PII_integral_n4_perv

Интеграл	Приближение интеграла	Абсолютная погрешность приближения	Параметры программы
$\int_0^{\pi/2} \cos(x) e^{\sin(x)} dx$	1.71828182845905E+0000	4.33680868994202E-0019	$n = 4, p = 1024$
$\int_0^{500} \cos(x) e^{\sin(x)} dx$	-3.73603552314934E-0001	1.89735380184963E-0019	$n = 4,$ $p = 1024000$
$\int_0^{\pi/2} \cos(x) dx$	1.00000000000000E+0000	1.08420217248550E-0019	$n = 4, p = 2048$
$\int_0^{\pi/2} \sqrt{1 - 2^{-1} \sin^2(x)} dx$	1.35064388104768E+0000	0.00000000000000E+0000	$n = 4, p = 64$
$\int_0^{2\pi} (e^{x/2} + \cos(4x)) dx$	4.42813852655585E+0001	1.38777878078145E-0017	$n = 4, p = 4096$
$\int_0^{2\pi} (x e^{-x} \cos(2x)) dx$	-1.22122604618968E-0001	2.50721752387273E-0019	$n = 4, p = 2048$

Сумму (35), аналогично (34), можно вычислять параллельно с использованием схемы сдваивания. С оговоркой относительно распараллеливания вычисления двойной суммы (35), с точностью до числового множителя сохранится оценка (21).

Замечание 4. Возможность представления (33) как непрерывного аналитического приближения первообразной подынтегральной функции является отличительной особенностью метода, обусловленной алгебраической формой полиномов с числовыми коэффициентами, приближающих подынтегральную функцию и интеграл. Известные методы аналогичную возможность непосредственно не представляют.

Целесообразно исследовать применимость непрерывного аналитического приближения первообразной для приближенного решения интегральных уравнений на основе кусочной интерполяции [10; 11], а также для приближенного решения дифференциальных уравнений на этой же основе [12]. Согласно эксперименту погрешность приближенного вычисления первообразной изложенным способом имеет такой же порядок, как погрешность вычисления интеграла (табл. 4).

Аналитические оценки погрешности кусочно-интерполяционного вычисления интеграла. Очевидно,

$$\left| \int_a^b f(x) dx - \int_a^b F_{L_n}(x) dx \right| \leq \int_a^b \max_{[a,b]} |f(x) - F_{L_n}(x)| dx = \int_a^b \max_{0 \leq i \leq p-1} \max_{[a_i, b_i]} |f(x) - F_{L_n}(x)| dx.$$

Отсюда

$$\left| \int_a^b f(x) dx - \int_a^b F_{L_n}(x) dx \right| \leq \max_{0 \leq i \leq p-1} \max_{[a_i, b_i]} |f(x) - F_{L_n}(x)| \times (b - a). \quad (42)$$

Согласно (23)

$$\max_{[a_i, b_i]} |f(x) - F_{L_n}(x)| = \max_{[a_i, b_i]} |f(x) - P_{L_{ni}}(t)|. \quad (43)$$

Погрешность интерполяции функции $f(x)$ полиномом Лагранжа на подынтервале оценивается следующим образом [1; 12]. При $k = \log_2 p$ справедливо неравенство

$$|f(x) - P_{L_{ni}}(t)| \leq c 2^{-k(n+1)} h^{n+1}, \quad p = 2^k, \quad x \in [a, b], \quad [a, b] = \bigcup_{i=0}^{2^k-1} [a_i, a_{i+1}]. \quad (44)$$

Лемма 2 [12]. Пусть для произвольного $n = \text{const}$ функция $f(x)$ определена, непрерывна и непрерывно дифференцируема $n + 1$ раз на отрезке $[a, b]$ из (13), на концах которого подразумеваются соответственные односторонние производные. Тогда, каково бы ни было $n \geq 1$, последовательность полиномов $P_{L_{ni}}(t)$ из (15) равномерно сходится к функции $f(x)$ на данном отрезке при $k \rightarrow \infty$, где $k = \log_2 p$, p – число подынтервалов из (13). Скорость сходимости оценивается из (44), где $c = \text{const}$, $h = (b - a) / n$ – шаг интерполяции полинома $P_{L_n}(t)$ из (9) на $[a, b]$ при $k = 0$.

При $n \leq n_0$, $n_0 = \text{const}$, имеет место

Следствие 1 [12]. В условиях леммы 2, а также в предположении $h < 1$, выполняется неравенство

$$|f(x) - P_{L_{ni}}(t)| \leq c 2^{-2k} h^2, \quad \forall x \in [a, b], \quad [a, b] = \bigcup_{i=0}^{2^k-1} [a_i, a_{i+1}], \quad (45)$$

где $c = \text{const}$ не зависит от выбора степени полинома n .

Подстановка (44) в (43) и соответственно в (42) влечет

Следствие 2. В условиях леммы 2 имеет место сходимость рассматриваемого приближения $\int_a^b F_{L_n}(x) dx$ к интегралу $\int_a^b f(x) dx$ при $k \rightarrow \infty$, для скорости сходимости верна оценка

$$\left| \int_a^b f(x) dx - \int_a^b F_{L_n}(x) dx \right| \leq c 2^{-k(n+1)} (b-a)^{n+2} / n^{n+1}. \quad (46)$$

Аналогичная подстановка (45) влечет

Следствие 3. В условиях следствия 1 также имеет место сходимость рассматриваемого приближения интеграла, для скорости сходимости верна оценка

$$\left| \int_a^b f(x) dx - \int_a^b F_{L_n}(x) dx \right| \leq c 2^{-2k} (b-a)^3 / n^2. \quad (47)$$

Следствия 2, 3 и оценки (46), (47) дополняют утверждения лемм 1, 2 и теоремы 2.

Рассмотренные условия и оценки уточняют аналоги, ранее приведенные в [1]. В частности, при $n = 2$ оценка (47) перейдет в оценку

$$\left| \int_a^b f(x) dx - \int_a^b F_{L_n}(x) dx \right| \leq c 2^{-2(k+1)} (b-a)^3,$$

которая верна при условии трехкратной дифференцируемости подынтегральной функции.

Кусочно-интерполяционное вычисление интеграла на основе полиномов Ньютона. При каждом i из (13) интерполяционный полином Ньютон а для интерполирования вперед с равноотстоящими узлами на i -м подынтервале имеет вид [5]

$$\Psi_{N_{ni}}(t) = f(x_{i0}) + \sum_{j=1}^n \frac{\Delta^j f_{i0}}{j!} \prod_{r=0}^{j-1} (t - r), \quad i \quad i \quad i \quad pn \quad pn$$

$$x \in [a_i, b_i], \quad t = (x - a_i) h_{pn}^{-1}, \quad h_{pn} = (b - a) / p / n, \quad i \in \overline{0, p-1}, \quad (48)$$

где узлы интерполяции $x_{ij} \in [a_i, b_i]$, $j \in \overline{0, n}$, $x_{i(j+1)} - x_{ij} = h_{pn}$, $h_{pn} = (b_i - a_i) n^{-1}$, $j \in \overline{0, n-1}$, $\Delta^j f_{i0}$ – конечная разность j -го порядка в узле $x_{i0} = a_i$, p из (13). Поскольку интерполяционные полиномы равной степени с одинаковыми узлами и узловыми значениями совпадают, то для последовательности полиномов $\Psi_{N_{ni}}(t) = P_{L_{ni}}(t)$, $t = (x - a_i) h_{pn}^{-1}$, верны оценка (44), лемма 2, следствие 1 и оценка (45), а последовательность полиномов (48) в условиях леммы 2 и следствия 1 сходится к $f(x)$ на $[a, b]$, если $p \rightarrow \infty$.

На каждом подынтервале $\Psi_{Nni}(t)$ можно преобразовать к виду алгебраического полинома с числовыми коэффициентами. Для этого каждый полином $\Psi_j(t) = \prod_{r=0}^{j-1} (t-r)$ из (48) преобразуется в полином с целочисленными коэффициентами. Если положить $t_r = r$, $r \in \overline{0, j-1}$, по схеме (2) получится

$$\Psi_j(t) = q_{0j} + q_{1j}t + q_{2j}t^2 + \dots + q_{jj}t^j, \quad (49)$$

где коэффициенты – целые числа. Почленное деление таких полиномов на $j!$ в (48) повлечет развернутую форму полинома вида

$$\Psi_{Nni}(t) = f(x_{i0}) + \sum_{j=1}^n \Delta^j f_{i0} (c_{0j} + c_{1j}t + c_{2j}t^2 + \dots + c_{jj}t^j), \quad i \in \overline{0, p-1}, \quad (50)$$

$$x_{i0} = a_i, \quad t = (x - a_i)h_{pn}^{-1}, \quad x \in [a_i, b_i],$$

$$h_{pn} = (b - a) / p / n, \quad i \in \overline{0, p-1}, \quad (50)$$

где

$$c_{\ell j} = q_{\ell j} / j!, \quad \ell \in \overline{0, j}, \quad j \in \overline{1, n}, \quad (51)$$

$q_{\ell j}$ из (49). С учетом $t = (x - a_i)h_{pn}^{-1}$ отсюда следует

$$\int_{a_i}^{b_i} \Psi_{Nni}(t) dx = h_{pn} \int_0^n f(x_{i0}) dt + \sum_{j=1}^n \Delta^j f_{i0} h_{pn} \int_0^n (c_{0j} + c_{1j}t + c_{2j}t^2 + \dots + c_{jj}t^j) dt.$$

Подстановка пределов интегрирования влечет

$$\int_{a_i}^{b_i} \Psi_{Nni}(t) dx = h_{pn} f(x_{i0}) t_0^n + h_{pn} \sum_{j=1}^n \Delta^j f_{i0} \left(c_{0j} t_0^n + c_{1j} \frac{1}{2} t_0^{2n} + c_{2j} \frac{1}{3} t_0^{3n} + \dots + c_{jj} \frac{1}{j+1} t_0^{(j+1)n} \right),$$

или

$$\int_{a_i}^{b_i} \Psi_{Nni}(t) dx = n h_{pn} f(x_{i0}) + h_{pn} \sum_{j=1}^n \Delta^j f_{i0} (c_{0j} n + 2^{-1} c_{1j} n^2 + 3^{-1} c_{2j} n^3 + \dots + (j+1)^{-1} c_{jj} n^{j+1}). \quad (52)$$

Сложение (52) по всем подынтервалам влечет формулу приближения интеграла. С учетом равенства интерполяционных полиномов Лагранжа и Ньютона в случае совпадения узлов и значений в узлах интерполяции имеет место

Теорема 3. На основе кусочной интерполяции подынтегральной функции с помощью полиномов Ньютона (48) значение интеграла можно приближенно вычислить из соотношения

$$\int_a^b f(x) dx \approx h_{pn} \sum_{i=0}^{p-1} \left(n f(x_{i0}) + \sum_{j=1}^n \Delta^j f_{i0} (c_{0j} n + 2^{-1} c_{1j} n^2 + 3^{-1} c_{2j} n^3 + \dots + (j+1)^{-1} c_{jj} n^{j+1}) \right),$$

$$h_{pn} = (b - a) / p / n. \quad (53)$$

где $f(x_{ij})$ – значение подынтегральной функции в j -м узле интерполяции на i -м подынтервале из (13), $\Delta^j f_{i0}$ – конечная разность j -го порядка в узле $x_{i0} = a_i$, p из (13), коэффициенты и числовые параметры определяются согласно (49), (50), (51). Погрешность приближения в условиях леммы 2 и следствий 1 – 3 оценивается, соответственно, из (46), (47) при соответственной замене обозначений.

Выражение в скобках внутренней суммы в (53) является числом, зависящим от степени интерполяционного полинома и от номера узла интерполяции, но не зависящим от номера подынтервала и вида подынтегральной функции. Ниже это число обозначается

$$c_{Nnj} = c_{0j} n + 2^{-1} c_{1j} n^2 + 3^{-1} c_{2j} n^3 + \dots + (j+1)^{-1} c_{jj} n^{j+1}. \quad (54)$$

Тогда (53) можно записать в виде

$$\int_a^b f(x) dx \approx h_{pn} \sum_{i=0}^{p-1} \left(n f(x_{i0}) + \sum_{j=1}^n \Delta^j f_{i0} c_{Nnj} \right), \quad h_{pn} = (b-a) / p / n. \quad (55)$$

Формулы (53), (55) могут непосредственно использоваться для вычисления интеграла. Коэффициенты c_{Nnj} априори вычисляются для всех функций и подынтервалов, записываются в память компьютера (в раздел описания констант программы) и в таком виде применяются для вычисления интегралов. Непосредственно ниже приводится таблица значений c_{Nnj} из (54), (55) для различных степеней полинома Ньютона.

Таблица 5

Значения коэффициентов c_{Nnj} / n вычисления интеграла из (55) для различных степеней n интерполяционного полинома Ньютона

n	$c_{Nnj} / n, j \in \overline{1, n}$
1	(5.00000000000000000000E-1)
2	(1.00000000000000000000E+0, 1.66666666666666666667E-1)
3	(1.50000000000000000000E+0, 7.50000000000000000000E-1, 1.25000000000000000000E-1)
4	(2.00000000000000000000E+0, 1.66666666666666666667E+0, 6.66666666666666666667E-1, 7.77777777777777777778E-2)
5	(2.50000000000000000000E+0, 2.91666666666666666667E+0, 1.87500000000000000000E+0, 5.90277777777777777778E-1, 6.59722222222222222222E-2)
6	(3.00000000000000000000E+0, 4.50000000000000000000E+0, 4.00000000000000000000E+0, 2.05000000000000000000E+0, 5.50000000000000000000E-1, 4.880952380952380952381E-2)
7	(3.50000000000000000000E+0, 6.41666666666666666667E+0, 7.29166666666666666667E+0, 5.18194444444444444444E+0, 2.23125000000000000000E+0, 5.112268518518518518519E-1, 4.346064814814814814815E-2)
8	(4.00000000000000000000E+0, 8.66666666666666666667E+0, 1.20000000000000000000E+1, 1.09111111111111111111E+1, 6.48888888888888888889E+0, 2.397883597883597883598E+0, 4.867724867724867724868E-1, 3.488536155202821869489E-2)
9	(4.50000000000000000000E+0, 1.12500000000000000000E+1, 1.83750000000000000000E+1, 2.03625000000000000000E+1, 1.54687500000000000000E+1, 7.897767857142857142857E+0, 2.565401785714285714286E+0, 4.62656250000000000000E-1, 3.188616071428571428571E-2)
10	(5.00000000000000000000E+0, 1.41666666666666666667E+1, 2.66666666666666666667E+1, 3.48611111111111111111E+1, 3.22500000000000000000E+1, 2.102843915343915343915E+1, 9.417989417989417989418E+0, 2.724316578483245149912E+0, 4.458774250440917107584E-1, 2.683414836192613970392E-2)

Программная реализация кусочно-интерполяционного вычисления интеграла на основе полинома Ньютона и соотношения (55) имеет следующий вид.

```

program PIN_integral_n1_10; {$APPTYPE CONSOLE} uses SysUtils; var a,b,S:extended; n:byte; p:int64;
function f(x:extended):extended;
begin f:=cos(x)*exp(sin(x)){cos(x)} {sqrt(1-0.5*sqrt(sin(x)))} {exp(x/2)+cos(4*x)} {x*exp(-x)*cos(2*x)} end;
function exact_int(a,b:extended):extended;
begin exact_int:=exp(sin(b))-exp(sin(a)){sin(b)-sin(a)} {1.35064388104767550252} {2*(exp(Pi)-1)}
{1/25*(exp(-2*Pi)*(3-10*Pi)-3)} end;
function PIN_int(a,b:extended; n:byte; p:int64):extended; type coeff = array[0..10] of extended;
const c:array[1..10,1..10] of extended =
((0.5, 0, 0, 0, 0, 0, 0, 0, 0, 0), (1,1.66666666666666666667E-1, 0, 0, 0, 0, 0, 0, 0, 0),
(1.5, 7.5E-1, 1.25E-1, 0, 0, 0, 0, 0, 0, 0), (2, 1.66666666666666666667E+0, 6.66666666666666666667E-1,
7.77777777777777777778E-2, 0, 0, 0, 0, 0, 0), (2.5, 2.91666666666666666667E+0, 1.875,
5.90277777777777777778E-1, 6.59722222222222222222E-2, 0, 0, 0, 0, 0), (3, 4.5, 4, 2.05, 5.5E-1,
4.880952380952380952381E-2, 0, 0, 0, 0), (3.5, 6.41666666666666666667E+0,
7.29166666666666666667E+0, 5.18194444444444444444E+0, 2.23125E+0,
5.112268518518518518519E-1, 4.346064814814814814815E-2, 0, 0, 0),
(4, 8.66666666666666666667E+0, 1.2E+1, 1.09111111111111111111E+1,
6.48888888888888888889E+0, 2.397883597883597883598E+0, 4.867724867724867724868E-1,
3.488536155202821869489E-2, 0, 0), (4.5, 1.125E+1, 1.8375E+1, 2.03625E+1, 1.546875E+1,
7.897767857142857142857E+0, 2.565401785714285714286E+0, 4.6265625E-1,
3.188616071428571428571E-2, 0), (5, 1.41666666666666666667E+1, 2.66666666666666666667E+1,
    
```

```

3.48611111111111111111111111111111E+1,3.225E+1,2.102843915343915343915E+1,9.417989417989417989418E+0,
2.724316578483245149912E+0,4.458774250440917107584E-1,2.683414836192613970392E-2);
var ai,hp,hpn:extended; j:byte; i:int64; dy0:coeff;
procedure Konech_Raznoct(n:byte;a00,h:extended; var dy0:coeff);
var r,k:byte; x:coeff; dy:array[0..10,0..10] of extended;
begin for r:=0 to n do x[r]:=a00+r*h; for r:=0 to n-1 do dy[1,r]:=f(x[r+1])-f(x[r]);
for k:=2 to n do for r:=0 to n-k do dy[k,r]:=dy[k-1,r+1]-dy[k-1,r]; for k:=1 to n do dy0[k]:=dy[k,0]; end;
begin hp:=(b-a)/p; hpn:=hp/n; ai:=a; i:=0;s:=0;
repeat
Konech_Raznoct(n,ai,hpn,dy0); s:=s + f(ai); for j:=1 to n do s:=s + dy0[j]*c[n,j]; inc(i); ai:=a+i*hp;
until i = p; PIN_int:=s*hp;
end;
begin
a:=0; b:=Pi/2; n:=6; p:=64; S:=PIN_int(a,b,n,p); writeln('S=',S,' Pogr=',abs(S-exact_int(a,b))); readln;
end.

```

Результаты численных экспериментов с программой PIN_integral_n1_10 приведены в табл. 6.

Таблица 6

Результаты численного эксперимента по приближению интеграла из соотношения (55)

Интеграл	Приближение интеграла	Абсолютная погрешность приближения	Параметры программы
$\int_0^{\pi/2} \cos(x) e^{\sin(x)} dx$	1.71828182845905E+0000	0.000000000000000E+0000	$n = 6, p = 64$
$\int_0^{500} \cos(x) e^{\sin(x)} dx$	-3.73603552314934E-0001	4.60785923306339E-0019	$n = 8, p = 4096$
$\int_0^{\pi/2} \cos(x) dx$	1.000000000000000E+0000	0.000000000000000E+0000	$n = 7, p = 32$
$\int_0^{\pi/2} \sqrt{1-2^{-1} \sin^2(x)} dx$	1.35064388104768E+0000	0.000000000000000E+0000	$n = 1, p = 16$
$\int_0^{2\pi} (e^{x/2} + \cos(4x)) dx$	4.42813852655585E+0001	0.000000000000000E+0000	$n = 5, p = 512$
$\int_0^{2\pi} (x e^{-x} \cos(2x)) dx$	-1.22122604618968E-0001	0.000000000000000E+0000	$n = 10, p = 512$

На основе интерполяционного полинома Ньютона можно также приблизить первообразную (22). Пусть рассматривается кусочно-интерполяционное приближение подынтегральной функции

$$f(x) \approx F_{N_n}(x) \quad \forall x \in [a, b],$$

где

$$\forall x \in [a, b], \forall i = 0, 1, \dots, p-1: \quad x \in [a_i, b_i] \Rightarrow F_{N_n}(x) = \Psi_{N_{ni}}(t), \quad t = (x - a_i)h_{pn}^{-1}, \quad (56)$$

$\Psi_{N_{ni}}(t)$ из (50). Пусть по-прежнему $f(x)$ непрерывна $\forall x \in [a, b]$ из (13). Тогда эта функция интегрируема, ее первообразная (22) непрерывна на $[a, b]$. Приближение подынтегральной функции $F_{N_n}(x)$ рассматривается как самостоятельная функция, определенная на всем $[a, b]$, при этом определяющими являются соотношения (56). Как и в случае полинома Лагранжа, начальный и конечный узел интерполяции полинома (50) всегда находятся соответственно на левой и правой границе подынтервала, поэтому при $i = 0, 1, \dots, p-2$

$$F_{N_n}(a_{i+1}) = \Psi_{N_{n(i+1)}}(t) \Big|_{x=a_{i+1}} = \Psi_{N_{n(i+1)}}(0) = f(a_{i+1}) = f(b_i) = \Psi_{N_{ni}}(n) = \Psi_{N_{ni}}(t) \Big|_{x=b_i} = F_{N_n}(b_i). \quad (57)$$

Согласно (56), (57) функция $F_{N_n}(x)$ непрерывна $\forall x \in [a, b]$, поэтому интегрируема на $[a, b]$. Ее первообразная,

$$\Phi_{N_n}(x) = \int_a^x F_{N_n}(x) dx + C, \tag{58}$$

$\forall C = \text{const}$ определена и непрерывна $\forall x \in [a, b]$. Первообразную (58) можно построить из полиномов (50). В случае $x \in [a_i, b_i]$ интеграл

$$\int_{a_i}^x \Psi_{N_{ni}}(t) dx = h_{pn} f(x_{i0}) \int_{a_i}^x dt + \sum_{j=1}^n \Delta^j f_{i0} h_{pn} \int_{a_i}^x (c_{0j} + c_{1j}t + c_{2j}t^2 + \dots + c_{jj}t^j) dt \tag{59}$$

представляет собой первообразную полинома (50). Равенство (59) эквивалентно

$$\int_{a_i}^x \Psi_{N_{ni}}(t) dx = h_{pn} f(x_{i0}) t \Big|_{a_i}^x + h_{pn} \left(c_{1j} t^2 \Big|_{a_i}^x + 2 c_{2j} t^3 \Big|_{a_i}^x + 3 c_{3j} t^4 \Big|_{a_i}^x + \dots + c_{jj} (j+1)^{-1} t^{j+1} \Big|_{a_i}^x \right),$$

$$x_{i0} = a_i, \quad t = (x - a_i) h_{pn}^{-1}, \quad x \in [a_i, b_i], \quad h_{pn} = (b - a) / p / n, \quad i \in \overline{0, p-1}, \tag{60}$$

Для полинома (60) вводится обозначение

$$\Psi_{INTN_{ni}}(x) = h_{pn} f(x_{i0}) t \Big|_{a_i}^x + h_{pn} \sum_{j=1}^n \Delta^j f_{i0} (c_{0j} t \Big|_{a_i}^x + 2^{-1} c_{1j} t^2 \Big|_{a_i}^x + 3^{-1} c_{2j} t^3 \Big|_{a_i}^x + \dots + c_{jj} (j+1)^{-1} t^{j+1} \Big|_{a_i}^x),$$

$$x \in [a_i, b_i]. \tag{61}$$

В этом обозначении

$$\int_{a_i}^x \Psi_{N_{ni}}(t) dx = \Psi_{INTN_{ni}}(x). \tag{62}$$

В скобках (60), (61) каждая нижняя подстановка равна нулю, поэтому

$$\int_{a_r}^{b_r} \Psi_{N_{nr}}(t) dx = \int_{a_r}^{a_{r+1}} \Psi_{N_{nr}}(t) dx = \Psi_{INTN_{nr}}(a_{r+1}) \quad \forall r = 0, 1, \dots, p-1. \tag{63}$$

Из (62), (63) следует

$$\int_{a_i}^x \Psi_{N_{ni}}(t) dx + \int_{a_{i-1}}^{a_i} \Psi_{N_{n(i-1)}}(t) dx = \Psi_{INTN_{ni}}(x) + \Psi_{INTN_{n(i-1)}}(a_i).$$

Отсюда с учетом (56)

$$\int_{a_i}^x F_{N_n}(x) dx + \int_{a_{i-1}}^{a_i} F_{N_n}(x) dx = \Psi_{INTN_{ni}}(x) + \Psi_{INTN_{n(i-1)}}(a_i),$$

или, с учетом (13),

$$\int_{a_{i-1}}^x F_{N_n}(x) dx = \Psi_{INTN_{ni}}(x) + \Psi_{INTN_{n(i-1)}}(b_{i-1}). \tag{64}$$

Аналогично, добавление к обеим частям (64) $\int_{a_{i-2}}^{a_{i-1}} \Psi_{N_{n(i-2)}}(t) dx$ влечет

$$\int_{a_i}^x \Psi_{N_{ni}}(t) dx + \int_{a_{i-1}}^{a_i} \Psi_{N_{n(i-1)}}(t) dx + \int_{a_{i-2}}^{a_{i-1}} \Psi_{N_{n(i-2)}}(t) dx = \Psi_{INTN_{ni}}(x) + \Psi_{INTN_{n(i-1)}}(a_i) + \Psi_{INTN_{n(i-2)}}(a_{i-1}).$$

С учетом определения (56) функции $F_{N_n}(x)$

$$\int_a^x F_{N_n}(x) dx + \int_{a_{i-1}}^{a_i} F_{N_n}(x) dx + \int_{a_{i-2}}^{a_{i-1}} F_{N_n}(x) dx = \Psi_{INT N_n i}(x) + \Psi_{INT N_n (i-1)}(a_i) + \Psi_{INT N_n (i-2)}(a_{i-1}),$$

или,

$$\int_{a_{i-2}}^x F_{N_n}(x) dx = \Psi_{INT N_n i}(x) + \Psi_{INT N_n (i-1)}(b_{i-1}) + \Psi_{INT N_n (i-2)}(b_{i-2}).$$

Процесс можно продолжать до получения нулевого индекса в нижнем пределе интегрирования. С учетом $a_0 = a$ получится

$$\int_a^x F_{N_n}(x) dx = \Psi_{INT N_n i}(x) + \Psi_{INT N_n (i-1)}(b_{i-1}) + \Psi_{INT N_n (i-2)}(b_{i-2}) + \dots + \Psi_{INT N_n (i-3)}(b_{i-3}) + \dots + \Psi_{INT N_n 0}(b_0), \quad x \in [a_i, b_i]. \quad (65)$$

В краткой записи

$$\int_a^x F_{N_n}(x) dx = \Psi_{INT N_n i}(x) + \sum_{\ell=1}^i \Psi_{INT N_n (i-\ell)}(b_{i-\ell}), \quad x \in [a_i, b_i]. \quad (66)$$

Таким образом, первообразная (58) кусочно-интерполяционного приближения (56) подынтегральной функции выражается из (65), (66). При этом $\Psi_{INT N_n r}(b_r)$ вычисляется из (61). Первообразная (58) аналитически и непрерывно приближает первообразную подынтегральной функции, если первообразная последней сама по себе аналитического выражения не имеет. При $x = b$ эти соотношения дают формулу приближенного вычисления интеграла

$$\int_a^b F_{N_n}(x) dx = \Psi_{INT N_n (p-1)}(b_{p-1}) + \Psi_{INT N_n (p-2)}(b_{p-2}) + \dots + \Psi_{INT N_n (i-1)}(b_{i-1}) + \Psi_{INT N_n (i-2)}(b_{i-2}) + \dots + \Psi_{INT N_n 1}(b_1) + \Psi_{INT N_n 0}(b_0), \quad b$$

точнее,

$$\int_a^b f(x) dx \approx \int_a^b F_{N_n}(x) dx = \sum_{i=0}^{p-1} \Psi_{INT N_n i}(b_i), \quad (67)$$

где $\Psi_{INT N_n i}(b_i)$ вычисляется из (61) при $x = b_i$.

В обозначении (62) первообразная (58) представляется в виде

$$\Phi_{N_n}(x) = \Psi_{INT N_n i}(x) + \Psi_{INT N_n (i-1)}(b_{i-1}) + \Psi_{INT N_n (i-2)}(b_{i-2}) + \dots + \Psi_{INT N_n 1}(b_1) + \Psi_{INT N_n 0}(b_0), \quad \forall x \in [a_i, b_i], \quad \forall i \in \overline{0, p-1}, \quad (68)$$

Лемма 3. На основе кусочной интерполяции подынтегральной функции с помощью полиномов Ньютона (48) первообразная (22) приближенно вычисляется из соотношения

$$\Phi(x) \approx \int_a^x F_{N_n}(x) dx = \Psi_{INT N_n i}(x) + \sum_{\ell=0}^{i-1} \Psi_{INT N_n \ell}(b_\ell), \quad \forall x \in [a_i, b_i], \quad \forall i \in \overline{0, p-1}, \quad (69)$$

где $\Psi_{INT N_n i}(x)$ определяется из (61), $[a_i, b_i]$ – из (13).

По основной теореме интегрального исчисления для (68), (69) должно выполняться равенство

$$\int_a^b F_{N_n}(x) dx = \Phi_{N_n}(b) - \Phi_{N_n}(a).$$

Аналогично (39),

$$\Phi_{N_n}(b) - \Phi_{N_n}(a) = \Phi_{N_n}(b) = \int_a^b F_{N_n}(x) dx = \sum_{i=0}^{p-1} \Psi_{INT N_n i}(b_i),$$

что также приводит к (67). Окончательно, имеет место

Теорема 4. В условиях леммы 3

$$\int_a^b f(x) dx \approx \Phi_{N_n}(b) = \int_a^b F_{N_n}(x) dx = \sum_{i=0}^{p-1} \Psi_{INT N_n i}(b_i), \tag{70}$$

где $F_{N_n}(x)$ определяется из (57), $\Psi_{INT N_n i}(b_i)$ вычисляется из (61) при $x = b_i$. На случай кусочной интерполяции подынтегральной функции с помощью полиномов Ньютона (48) дословно переносятся лемма 2, следствия 1 – 3 при условии замены в этих утверждениях $P_{L_n i}(t)$ из (15) на $\Psi_{N_n i}(t)$ из (50), (51) и $F_{L_n}(x)$ из (23) на $F_{N_n}(x)$ из (56). При $n = 2$ оценка (47) перейдет в оценку

$$\left| \int_a^b f(x) dx - \int_a^b F_{N_n}(x) dx \right| \leq c 2^{-2(k+1)} (b-a)^3,$$

которая верна при условии трехкратной дифференцируемости подынтегральной функции.

Коэффициенты полинома $\Psi_{N_n i}(t)$ из (50), $c_{\ell j} = q_{\ell j} / j!$, $\ell \in \overline{0}, j, j \in \overline{1}, n$, где $q_{\ell j}$ из (49), не зависят от подынтегральной функции, от номера подынтервала, от диапазона независимой переменной, они зависят от степени полинома, номера коэффициента и от номера узла интерполяции. Эти коэффициенты можно вычислить априори для всех функций и подынтервалов, записать в память компьютера (в раздел описания констант программы) и хранить для вычисления интегралов рассматриваемого вида. В качестве примера ниже приводится таблица значений коэффициентов $c_{\ell j}$ для интерполяционного полинома Ньютона степени $n = 10$.

Таблица 7

Значения коэффициентов $c_{\ell j}$ для полинома Ньютона степени $n = 10$

j	$c_{\ell j}, \ell \in \overline{0}, j$
1	(0, 5.00000000000000000000E-1)
2	(0, -2.50000000000000000000E-1, 1.66666666666666666666E-1)
3	(0, 1.66666666666666666666E-1, -1.66666666666666666666E-1, 4.16666666666666666666E-2)
4	(0, -1.25000000000000000000E-1, 1.52777777777777777777E-1, -6.25000000000000000000E-2, 8.33333333333333333333E-3)
5	(0, 1.00000000000000000000E-1, -1.38888888888888888888E-1, 7.29166666666666666666E-2, -1.66666666666666666666E-2, 1.38888888888888888888E-3)
6	(0, -8.33333333333333333333E-2, 1.26851851851851851851E-1, -7.81250000000000000000E-2, 2.36111111111111111111E-2, -3.47222222222222222222E-3, 1.98412698412698412698E-4)
7	(0, 7.14285714285714285714E-2, -1.16666666666666666666E-1, 8.05555555555555555555E-2, -2.91666666666666666666E-2, 5.78703703703703703703E-3, -5.95238095238095238095E-4, 2.48015873015873015873E-5)
8	(0, -6.25000000000000000000E-2, 1.08035714285714285714E-1, -8.14236111111111111111E-2, 3.35763888888888888888E-2, -8.10185185185185185185E-3, 1.14087301587301587301E-3, -8.68055555555555555555E-5, 2.755731922398589065256E-6)
9	(0, 5.55555555555555555555E-2, -1.006613756613756613757E-1, 8.13795194003527336860E-2, -3.70833333333333333333E-2, 1.031057098765432098765E-2, -1.78571428571428571428E-3, 1.88078703703703703703E-4, -1.102292768959435626102E-5, 2.755731922398589065256E-7)
10	(0, -5.00000000000000000000E-2, 9.429894179894179894180E-2, -8.07911706349206349206E-2, 3.988536155202821869489E-2, -1.23697916666666666666E-2, 2.49090608465608465608E-3, -3.25520833333333333333E-4, 2.663874191651969429747E-5, -1.24007936507936507936E-6, 2.505210838544171877505E-8)

При вычислении полинома (50) может использоваться схема (8) для внутренней суммы, затем вычисляется внешняя сумма парных произведений. Вычисление первообразной (65), (66) для $x \in [a_i, b_i]$ сводится к сложению значений полинома (50) с суммой значений аналогичных полиномов на правых границах предшествующих подынтервалов. В последовательном варианте вычислений можно переходить от подынтервала к подынтервалу, добавляя к текущему значению $\Psi_{INT N n i}(b_i)$ по одному слагаемому $\Psi_{INT N n (i-1)}(b_{i-1})$ с границы предшествующего подынтервала. При $x = b$ получится значение определенного интеграла.

Замечание 5. Для рассматриваемого варианта можно повторить замечание 2: на последнем подынтервале надо заново рассчитать шаг интерполяции $h_{pn} = (b - a_{p-1}) / n$, взяв точные границы подынтервала.

Программная реализация вычисления интеграла на основе полинома Ньютона с коэффициентами из табл. 7 имеет следующий вид.

```

program PIN_integral_n10_perv; {$APPTYPE CONSOLE} uses SysUtils; var a,b,S:extended; p:int64;
function f(x:extended):extended;
begin f:=cos(x)*exp(sin(x)){cos(x)}{sqrt(1-0.5*sqr(sin(x)))}{exp(x/2)+cos(4*x)}{x*exp(-x)*cos(2*x)} end;
function exact_int(a,b:extended):extended;
begin exact_int:=exp(sin(b))-exp(sin(a)){sin(b)-sin(a)}{1.35064388104767550252}{2*(exp(Pi)-1)}
{1/25*(exp(-2*Pi)*(3-10*Pi)-3)} end;
function PIN_int(a,b:extended; p:int64):extended; const n=10;
Cint:array[1..n,0..n] of extended =(0, 5.00000000000000000000E-1, 0, 0, 0, 0, 0, 0, 0, 0, 0),
(0, -2.50000000000000000000E-1, 1.66666666666666666666E-1, 0, 0, 0, 0, 0, 0, 0, 0),
(0, 1.66666666666666666666E-1, -1.66666666666666666666E-1, 4.16666666666666666666E-2, 0, 0, 0, 0, 0, 0, 0),
(0, -1.25000000000000000000E-1, 1.52777777777777777777E-1, -6.25000000000000000000E-2, 8.33333333333333333333E-3, 0, 0, 0, 0, 0, 0),
(0, 1.00000000000000000000E-1, -1.38888888888888888888E-1, 7.29166666666666666666E-2, -1.66666666666666666666E-2, 1.38888888888888888888E-3, 0, 0, 0, 0, 0),
(0, -8.33333333333333333333E-2, 1.26851851851851851851E-1, -7.81250000000000000000E-2, 2.36111111111111111111E-2, -3.47222222222222222222E-3, 1.98412698412698412698E-4, 0, 0, 0, 0, 0),
(0, 7.14285714285714285714E-2, -1.16666666666666666666E-1, 8.05555555555555555555E-2, -2.91666666666666666666E-2, 5.78703703703703703703E-3, -5.95238095238095238095E-4, 2.48015873015873015873E-5, 0, 0, 0, 0),
(0, -6.25000000000000000000E-2, 1.08035714285714285714E-1, -8.14236111111111111111E-2, 3.35763888888888888888E-2, -8.10185185185185185185E-3, 1.14087301587301587301E-3, -8.68055555555555555555E-5, 2.755731922398589065256E-6, 0, 0, 0),
(0, 5.55555555555555555555E-2, -1.006613756613756613757E-1, 8.137951940035273368607E-2, -3.70833333333333333333E-2, 1.031057098765432098765E-2, -1.78571428571428571428E-3, 1.88078703703703703703E-4, -1.102292768959435626102E-5, 2.755731922398589065256E-7, 0, 0),
(0, -5.00000000000000000000E-2, 9.42989417989417989418E-2, -8.07911706349206349206E-2, 3.988536155202821869489E-2, -1.23697916666666666666E-2, 2.49090608465608465608E-3, -3.25520833333333333333E-4, 2.663874191651969429747E-5, -1.24007936507936507936E-6, 2.505210838544171877505E-8);
type coeff = array[0..10] of extended; var ai,hp,hpn,s,pp2:extended; j,r:byte; i:int64; c:array[0..n] of extended;
dy0:coeff;
procedure Konech_Raznoct(n:byte;a00,h:extended; var dy0:coeff);
var r,k:byte; x:coeff; dy:array[0..10,0..10] of extended;
begin for r:=0 to n do x[r]:=a00+r*h; for r:=0 to n-1 do dy[1,r]:=f(x[r+1])-f(x[r]);
for k:=2 to n do for r:=0 to n-k do dy[k,r]:=dy[k-1,r+1]-dy[k-1,r]; for k:=1 to n do dy0[k]:=dy[k,0]; end;
begin
for j:=1 to n do begin pp2:= Cint[j][n]; for r:=1 to n do pp2 := pp2*n + Cint[j][n-r]; c[j]:=pp2; end;
hp:=(b-a)/p; hpn:=hp/n; i:=0; ai:=a; s:=0;
repeat
Konech_Raznoct(n,ai,hpn,dy0); s:= s + f(ai); for j:=1 to n do s:= s + dy0[j]*c[j]; inc(i); ai:=a+i*hp;
until i = p; PIN_int:=s*hp;
end;
begin
a:=0; b:=Pi/2; p:=8; S:=PIN_int(a, b, p); writeln('S=', S, ' Pogr=', abs(S-exact_int(a,b))); readln;
end.

```

Результаты работы программы представлены непосредственно ниже.

Таблица 8

Результаты численного эксперимента по приближению интеграла с помощью программы PIN_integral_n10_perp

Интеграл	Приближение интеграла	Абсолютная погрешность приближения	Параметры программы
$\int_0^{\pi/2} \cos(x) e^{\sin(x)} dx$	1.71828182845905E+0000	2.16840434497101E-0019	$n = 10, p = 8$
$\int_0^{500} \cos(x) e^{\sin(x)} dx$	-3.73603552314934E-0001	7.31836466427715E-0019	$n = 10, p = 2000$
$\int_0^{\pi/2} \cos(x) dx$	1.00000000000000E+0000	0.00000000000000E+0000	$n = 10, p = 16$
$\int_0^{\pi/2} \sqrt{1 - 2^{-1} \sin^2(x)} dx$	1.35064388104768E+0000	1.08420217248550E-0019	$n = 10, p = 16$
$\int_0^{2\pi} (e^{x/2} + \cos(4x)) dx$	4.42813852655585E+0001	0.00000000000000E+0000	$n = 10, p = 8$
$\int_0^{2\pi} (x e^{-x} \cos(2x)) dx$	-1.22122604618968E-0001	0.00000000000000E+0000	$n = 10, p = 512$

Согласно эксперименту погрешность приближения первообразной изложенным способом имеет такой же порядок, как погрешность вычисления интеграла в табл. 8.

Обсуждение вариантов приближения интегралов, первообразных и стандартизации программ. Как видно из таблиц 2 и 6, приближение интеграла на основе полинома Лагранжа из соотношения (19) оказывается несколько точнее приближения на основе полинома Ньютона из (55). При этом оба приближения сравнительно точны и согласно эксперименту отличаются стабильностью. Кроме того, оба эти приближения превосходят по точности аналоги из [1], особенно положительно в данном отношении отличается приближение на основе полинома Лагранжа из (19). Данное приближение, кроме того, положительно отличается по трудоемкости, в частности, от методов вычисления интеграла, представленных в [6]. Напротив, сопоставление таблиц 4 и 8 показывает, что приближение на основе полинома Ньютона из соотношения (70) точнее приближения на основе полинома Лагранжа из (36). То же относится к приближениям первообразных, которые в обоих случаях являются аналитическими и непрерывными, что характерно отличает метод от известных [2; 6; 13]. По структуре отличия предложенного метода сохраняется относительно современных моди-

фикаций формул Ньютона-Котеса [14; 15]. Кроме того, согласно таблицам 2, 6 и на основании данных численных экспериментов, описанных в [2; 6], предложенный метод по отношению к существующим обладает потенциалом повышенной точности приближения. Однако для корректности такого сравнения нужна статистическая обработка данных, которая в изложенной выше работе не выполнялась. Необходимо отметить, что приближения полиномиального вида из соотношений (36) и (70) уступают в точности приближениям на основе линейных комбинаций из (19) и (55). Это объясняется тем, что вычисление полинома по схеме Горнера накапливает погрешность с ростом степени полинома больше, чем вычисление линейной комбинации с постоянными коэффициентами. По этой причине в табл. 4 пришлось ограничить эксперимент полиномом степени $n = 4$, а в табл. 8 – полиномом степени $n = 10$. Все рассмотренные варианты приближений интеграла являются сравнительно быстродействующими и обладают максимальным параллелизмом.

Программы, реализующие приближения, достаточно компактны, коэффициенты приближений в них являются хранимыми в разделах описания констант, отсюда сравнительное быстродействие их выполнения. Для запуска соответствующей программы пользователю достаточно задать на ее вхо-

де вид подынтегральной функции, промежуток интегрирования, степень полинома и число подынтервалов в качестве параметров. Можно упростить пользовательский интерфейс, если дополнить программу выбором наилучшего приближения подынтегральной функции. Автоматическая реализация такого приближения элементарно выполняется в [1]. В частности, оно осуществимо для наиболее высокой точности приближения путем соответствующего подбора степени n и числа подынтервалов p . В случае если программа таким образом модифицирована, пользователю достаточно задавать на входе только вид подынтегральной функции и границы промежутка интегрирования. Следует, однако, принять во внимание, что согласно практике компьютерных вычислений наилучшее приближение подынтегральной функции не означает высшую точность вычисления интеграла.

В заключение относительно точности приближения первообразной можно отметить следующее. Номер подынтервала, при котором $x \in [a_i, b_i]$, есть $i = [(x - a) / ((b - a)p^{-1})]$. В соотношении (69)

$$\sum_{\ell=0}^{i-1} \Psi_{INT N n \ell}(b_{\ell}) \text{ совпадает с интегралом} \\ \int_a^{b_{i-1}} F_{N n}(x) dx. \text{ Для повышения точности}$$

приближения первообразной этот интеграл можно вычислить из (55) с заменой b на b_{i-1} , и только на i -м подынтервале вычислить $\Psi_{INT N n i}(x)$, затем сложить с полученным интегралом. Аналогично можно действовать при вычислении (33), (34).

Заключение

Компьютерное вычисление интеграла от функции одной вещественной переменной выполняется на основе кусочно-интерполяционного приближения подынтегральной функции с помощью интерполяционных полиномов Лагранжа и Ньютона, преобразованных в каноническую форму алгебраических полиномов с числовыми коэффициентами. Коэффициенты интерполяционных полиномов записываются в память компьютера (в раздел описания констант программы), они инвариантны относительно вида подынтегральной функции и промежутка интегрирования. Фактически в работе выполнена разновидность реализации формул Ньютона-Котеса на основе кусочной интерполяции, при этом достигается сравнительно высокая точность приближения интеграла, в том числе на больших промежутках

интегрирования, и стандартизируется пользовательский интерфейс программ, реализующих метод. Метод распространяется на непрерывное аналитическое компьютерное приближение первообразной подынтегральной функции. Выполнены численные эксперименты, показывающие стабильность метода для различных степеней полиномов, в частности достигается граница абсолютной погрешности вычисления интеграла на промежутке $[a, b] = [0, 500]$ порядка 10^{-20} . При этом на практике достигается минимизация погрешности и временной сложности кусочно-интерполяционного вычисления интегралов, а также стандартизация программ, реализующих метод.

Список литературы

1. Ромм Я.Е., Джанунц Г.А. Кусочная интерполяция функций, производных и интегралов с приложением к решению обыкновенных дифференциальных уравнений // Современные наукоемкие технологии. 2020. № 12-2. С. 291-316. DOI: 10.17513/snt.38448.
2. Quarteroni A., Sacco R., Saleri F. Numerical Mathematics. Berlin/Heidelberg: Springer, 2007. 656 p. DOI: 10.1007/b98885.
3. Ромм Я.Е. Локализация и устойчивое вычисление нулей многочлена на основе сортировки. II // Кибернетика и системный анализ. 2007. № 2. С. 161-174.
4. Березин И.С., Жидков Н.П. Методы вычислений. Т.2. М.: Наука, 1962. 640 с.
5. Березин И.С., Жидков Н.П. Методы вычислений. Т.1. М.: Наука, 1966. 632 с.
6. Burg C.O.E., Degny E. Derivative-based midpoint quadrature rule. Applied Mathematics. 2013. Vol. 4. P. 228-234. DOI: 10.4236/am.2013.41A035.
7. Wolfram|Alpha. [Электронный ресурс]. URL: <http://www.wolframalpha.com/> (дата обращения: 20.03.2023).
8. Волосова А.В. Параллельные методы и алгоритмы. Москва: МАДИ, 2020. 176 с.
9. Фихтенгольц Г.М. Курс дифференциального и интегрального исчисления. Т.2. Санкт-Петербург, Москва, Краснодар: Лань, 2020. 800 с.
10. Ромм Я.Е., Джанунц Г.А. Варьируемое кусочно-интерполяционное решение задачи Коши для уравнения переноса с итерационным уточнением // Современные наукоемкие технологии. 2020. № 1. С. 21-46. DOI: 10.17513/snt.37897.
11. Васильева А.Б., Тихонов Н.А. Интегральные уравнения. М.: Физматлит, 2002. 160 с.
12. Джанунц Г.А., Ромм Я.Е. Варьируемое кусочно-интерполяционное решение задачи Коши для обыкновенных дифференциальных уравнений с итерационным уточнением // Журнал вычислительной математики и математической физики. 2017. Т. 57. № 10. С. 1641-1660.
13. Zadorin A.I. Lagrange interpolation and Newton-Cotes formulas for functions with boundary layer components on piecewise-uniform grids. Numer. Anal. Appl. 8(3). 2015. P 235-247. DOI: 10.1134/S1995423915030040.
14. Zadorin N.A. Optimization of nodes of composite quadrature formulas in the presence of a boundary layer. Sib. Elektron. Mat. Izv. 18 (2). 2021. P. 1201-1209. DOI: 10.33048/semi.2021.18.091.
15. Задорин А.И., Задорин Н.А. Интерполяция Лагранжа и формулы Ньютона-Котеса на сетке Бахвалова при наличии пограничного слоя // Журнал вычислительной математики и математической физики. 2022. Т. 62. № 3. С. 355-366.