ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал ордена Трудового Красного Знамени
федерального государственного бюджетного образовательного учреждения
высшего образования
«Московский технический университет связи и информатики»

С.А. ШВИДЧЕНКО

Методические указания
для проведения лабораторных работ и практических занятий
по дисциплине

«ИНФОРМАТИКА (СПЕЦГЛАВЫ)»

Кафедра **«Информатика и вычислительная техника»**

Направление подготовки **11.03.02. Инфокоммуникационные технологии и системы связи**
Профили: **Многоканальные телекоммуникационные системы, Сети связи и системы коммутации, Системы радиосвязи и радиодоступа, Защищенные системы и сети связи**

Разработала:
Доцент кафедры ИВТ Швидченко С.А.

Ростов-на-Дону
2019

Методические указания
для проведения практических занятий
по дисциплине
«Информатика (спецглавы)»

Составитель: Швидченко С.А., доц. каф. «ИВТ»

Рассмотрено и одобрено
на заседании кафедры «ИВТ»
Протокол от «26» августа 2019 г., № 1.

Практическое занятие №1. Уточнение корня нелинейного уравнения методом Ньютона средствами пакета Excel и средствами пакета MathCad.

1. Условие задания

Применяя численные методы найти с точностью $\varepsilon=10^{-3}$ два корня нелинейного уравнения $F_2(x)=0$, где $F_2(x)$ – заданная функция квадратичной аппроксимации набора некоторых экспериментальных отсчётов. В процессе решения уравнения необходимо выполнить все пункты, указанные в задании курсовой работы.

$$F_2(x) = -0.0255x^2 + 0,1932x + 0,5059.$$

Для решения такого уравнения существует известная из школьного курса математики простая формула $X=(-b\pm\sqrt{b^2-4ac})/2a$, результатом вычислений по которой являются корни x_1= -2,06787 и x_2=9.78787. Это оказалось возможным только при положительном дискриминанте *D=b2–4ac*. В противном случае уравнение могло иметь только одно или ни одного решения.

Однако для решения алгебраических степени более 4-й и всех трансцендентных уравнений подобных формул не существует. Поэтому данные корни могут служить в этом задании только для контроля верности расчетов численными методами.

2. Отделение корней нелинейного уравнения.

Под отделением понимают процесс установления минимально возможного отрезка [n, k] оси X, на котором содержится только один корень уравнения $F_2(x)=0$. Этот процесс (метод) может быть графическим или аналитическим.

При графическом методе значениями корней заданного уравнения являются (см. рис.1) точки пересечения графика функции $F_2(x)$ с осью 0X. Полагая, что данные точки лежат внутри отрезков [n, k], можно оценить и произвольно выбрать начальные (x=n) и конечные (x=k) граничные значения интервалов изоляции для каждого из корней.

На рис.1 расположение точек таково, что отделить можно только один корень, который существует на отрезке [-3, -2]. Второй корень на этом рисунке увидеть невозможно. Поэтому следует достроить график функции вправо до пересечения с осью Y. Для этого необходимо реализовать процедуру табуляции (генерацию значений функции при изменении аргумента с фиксированным шагом). Шаг табуляции выбирается произвольно, исходя из желаемой точности определения отрезка. Таблица таких значений при единичном шаге аргумента (h=1) имеет следующий вид:

Таблица значений функции $F_2(x)=-0,0255x^2 + 0,1932x + 0,5059$

x	5	6	7	8	9	10	11
$F_2(x)$	0,8344	0,7471	0,6088	0,4195	0,1792	-0,1121	-0,4544

График функции представлен на рис.1. Легко видеть, что второй отрезок ограничен координатами [9, 10].

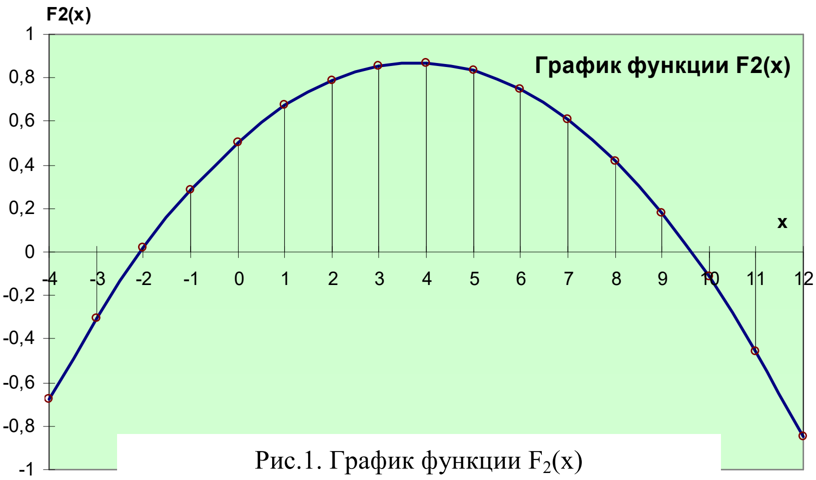


Рис.1. График функции $F_2(x)$

При аналитическом методе для отделения корней целесообразно воспользоваться набором числовых результатов табуляции функции f(x) при изменении аргумента x с шагом h и выбрать из них отрезки [n, k], в начале n и конце k которых непрерывная функция f(x) будет принимать значения разных знаков.

Аналитический способ использует теорему из математического анализа: если непрерывная функция f(x) принимает значения разных знаков на концах отрезка [k,m], то есть $f(k)*F(m)<0$, то внутри этого отрезка содержится, по меньшей мере, один корень уравнения f(x). Корень будет единственным, если производная f'(x) существует и сохраняет постоянный знак внутри интервала[k,m].

Таблица знаков функции $F_2(x)$:

x	-3	-2	9	10
$F_2(x)$	-0,3032	0,0175	0,1792	-0,1121
sign $F_2(x)$	-	+	+	-

На отрезке [-3, -2] и на отрезке [9, 10] функция меняет знак, следовательно, существует, по крайней мере, по одному корню на отрезке.

Первая производная функции

$$F_2'(x) = 2*(-0{,}0255x) + 0{,}1932;$$

вторая производная функции

$$F_2''(x) = -0{,}0510<0.$$

Знак второй производной функции свидетельствует о том, что первая ее производная является функцией монотонно убывающей, а сама функция является выпуклой. Монотонность следует из неизменного значения второй производной.

Таблица знаков производной функции:

x	-3	-2	9	10
$F_2'(x)$	0,3462	0,2952	-0,2658	-0,3168
sign $F_2(x)$	+	+	-	-

На отрезке [-3, -2] первая производная сохраняет знак и положительна, следовательно, функция $F_2(x)$ монотонно возрастает и имеет единственный корень на отрезке. На отрезке [9, 10] первая производная сохраняет знак и отрицательна, следовательно, функция монотонно убывает, и на отрезке также имеется единственный корень. Эти выводы полностью совпадают с графическим представлением функции.

Для уточнения первого корня используем метод простой итерации, а второго корня – метод Ньютона.

Контрольные вопросы ПЗ1(ОПК-4):

1. В чём заключается задача численного интегрирования?
2. Как группируются численные методы интегрирования?
3. Методы Ньютона – Котеса. Формулы прямоугольников?
4. Методы Ньютона – Котеса. Формула трапеций?
5. Методы Ньютона – Котеса. Формула Симпсона?

Практическое занятие №2. Структурные схемы алгоритмов, программирование и решение задач на ПК, связанных с решением нелинейных уравнений методами половинного деления, итерации и касательных (Ньютона – Рафсона).

Уточнение корня уравнения методом Ньютона

Метод Ньютона считается более практичным по сравнению с методом итерации, так как не требует отыскивать сходящуюся итерирующую функцию φ(x). Для уточнения на отрезке [9; 10] второго корня уравнения $F_2(x)$ этим методом с точностью ε используют рекуррентную формулу

$$x_{n+1} = x_n - \frac{F_2(x_n)}{F_2'(x_n)},\ n = 0,1,2,...$$

Начальное значение переменной x должно находиться в интервале отделенного отрезка $x_0 \in [n;\ k]$. Но при этом необходимо обеспечить условия сходимости итерационного процесса:

- функция $F_2(x)$ lдолжна быть непрерывна на отрезке локализации и иметь разные знаки на концах отрезка;

- первая и вторая производные функции отличны от нуля и сохраняют знак на отрезке. Эти условия выполняются, они проанализированы в предыдущих разделах.

Обычно из интервала локализации корня выбирают ту граничную точку $x_0 = n$ или $x_0 = k$, которая позволяет обеспечить условие сходимости вычислительного процесса

$$F_2(x_0) * F_2''(x_0) > 0$$

Например, при $x_0=10$:
$F_2(10) = -0,0255*10^2 +0,1932*10 + 0,5059 = -0,1121<0$;
$F''_2(10) = -0,0510<0$;

$$-0,1121 * (-0,0510) >0.$$

Таким образом, итерацию можно начинать от начального значения $x_0=10$. Кстати, вторая точка отрезка k=9 этому условию не удовлетворяет.

Вычислительный процесс последовательных приближений от начальной точки к истинному значению корня должен продолжаться до обеспечения точности его вычисления. При этом задаваемая точность ε обеспечивается при выполнении условия:

$$T = |X_n - X_{n-1}| \le \sqrt{\frac{2m_1\varepsilon}{M_2}},$$

где m_1=min | F2'(x) | и M_2=max | $F_2''(x)$ | на отрезке [n; k]. Но так как для исходного уравнения выполняется соотношение $2m_1 /M_2 >10^{-2}$, то вместо указанного условия можно использовать более простое [2], [3]:

$T = | X_{n+1} – X_n | = | F_2(X_n) / F_2'(X_n) | \le 0,1*\varepsilon$

Схема алгоритма уточнения корней по методу Ньютона представлена на рис.3.

Результаты программного решения уравнения $F_2(x) = 0$ на отрезке [9, 10] (см. приложение) приведены в таблице:

n	x_n	$F_2(x_n)$	$F_2'(x_n)$
0	10,0	-0,11210	-0,31680
1	9,6461	-0,00319	-0,29875
2	9,6355	Искомый корень	

X* - уточненный корень на отрезке [9, 10] равен 9,6355.

Погрешность вычисления методом Ньютона оценивается по формуле:

$$|x^* - x_n| \le \frac{M_2}{2m_1}(x_n - x_{n-1})^2 \approx 0,2 \cdot 10^{-2}$$

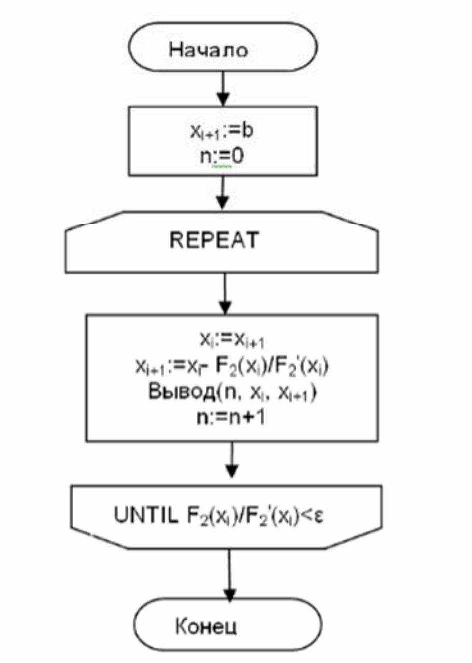


Рис.3. Схема алгоритма уточнения корней по методу Ньютона

Контрольные вопросы ПЗ2(ОПК-4):

1. Какое уравнение называется нелинейным?
2. Какое уравнение называется алгебраическим, а какое – трансцендентным?
3. Что является корнем нелинейного уравнения?
4. Из каких этапов состоит решение нелинейного уравнения?
5. Как производится этап отделения корней?
6. На какой теореме основывается аналитическое отделение корней?
7. При каких условиях корень будет единственным на отрезке отделения корней?
8. В чём заключается этап уточнения корней?
9. В чём суть метода половинного деления?
10. Что является условием окончания поиска корня?

11. В чём суть метода простой итерации уточнения корня?
12. Какие условия должны быть выполнены, чтобы процесс сходился к корню?
13. Приведите один из способов равносильного преобразования уравнения *f(x)=0*к виду *x= φ(x)*.
14. Что является условием окончания поиска корня в методе простой итерации?
15. В чём суть метода Ньютона - Рафсона уточнения корня?
16. Сравните методы уточнения корней.
17. Приведите структурную схему алгоритма заданного метода, уточняющего корень нелинейного уравнения.

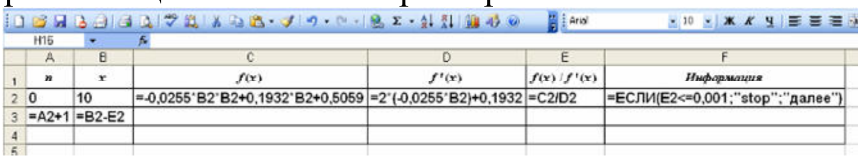
Практическое занятие №3. Метод наименьших квадратов. Структурные схемы алгоритмов, программирование и решение задач на ПК, связанных с решением задач аппроксимации методом наименьших квадратов.

Уточнение корня уравнения методом Ньютона средствами пакета Excel

Уточнение корня методом Ньютона средствами пакета Excel осуществляется по тем же принципам, что и расчет методом простой итерации в предыдущем разделе. Необходимо организовать циклический процесс расчета с возможностью многократного увеличения числа повторов тела цикла. Повторы реализуются копированием расчетных формул с визуальным контролем достижения требуемой точности.

Для этого требуется подготовить шапку таблицы и по расчетным формулам первую строку с начальными значениями переменных (A2:F2). Для рассматриваемой функции начальное значение x_0=10 (ячейка B2). В ячейке E2 значение соответствует части расчетной формулы, но одновременно и погрешности вычисления. Поэтому именно она применяется для визуального контроля достижения заданной точности (см.рис.5).

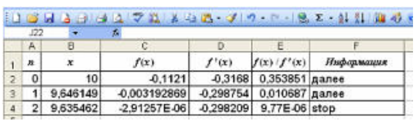
Вторая строка отличается тем, что вычисляет порядковый номер и рассчитывает последующее значение аргумента x. Остальные формулы копируются из первой строки. После формирования строки A3:F3, можно выполнять копирование целиком всей строки расчетов.



	A	B	C	D	E	F
1	n	x	f(x)	f '(x)	f(x) / f '(x)	Информация
2	0	10	=-0,0255*B2*B2+0,1932*B2+0,5059	=2*(-0,0255*B2)+0,1932	=C2/D2	=ЕСЛИ(E2<=0,001;"stop";"далее")
3	=A2+1	=B2-E2				
4						

Рис.5. Формулы для уточнения корня по методу Ньютона

Выполнение расчетов дает результаты, представленные на рисунке 6. По таблице видно, что сигнал "stop" выдается уже на третьем шаге уточнения. Это видно по ячейке E4, в которой ошибка значительно меньше допустимой. Значение переменной, при которой это достижимо показано в ячейке A4, это и есть приближенное значение корня. Таким образом, второй корень уравнения равен 9,6355.



	A	B	C	D	E	F
1	n	x	f(x)	f '(x)	f(x) / f '(x)	Информация
2	0	10	-0,1121	-0,3168	0,353851	далее
3	1	9,646149	-0,003192869	-0,298754	0,010687	далее
4	2	9,635462	-2,91257E-06	-0,298209	9,77E-06	stop

Рис.6. Результаты уточнения корня по методу Ньютона

Уточнение корня уравнения методом Ньютона средствами пакета MathCad

Для решения нелинейных уравнений вида f(x)=0 в Mathcad используется функция

root(f(x1, x2, …), x1, a, b).

Функция возвращает значение x1, принадлежащее отрезку [a, b], при котором выражение или функция f(x) обращается в 0. Оба аргумента этой функции должны быть скалярами. Функция возвращает скаляр. Функция ***root*** реализует вычисление корня уравнения численным методом (секущих прямых) с точностью ***TOL*** (по умолчанию ***TOL*** $=1{,}10^{-3}$).

Аргументы:

- f(x1, x2, …) - функция, определенная где-либо в рабочем документе, или выражение. Выражение должно возвращать скалярные значения;

- x1 - - имя переменной, которая используется в выражении. Этой переменной перед использованием функции ***root*** необходимо присвоить числовое значение (начальное приближение при поиске корня);

- a, b - необязательны, если используются, то должны быть вещественными числами, причем a < b.

Для изменения точности, с которой функция r***oot*** ищет корень, нужно изменить значение системной переменной ***TOL.*** Если значение ***TOL*** увеличивается, функция root будет сходиться быстрее, но ответ будет менее точен. Если значение ***TOL*** уменьшается, то функция ***root*** будет сходиться медленнее, но ответ будет более точен. Чтобы изменить значение ***TOL*** в определенной точке рабочего документа, используйте определение вида. Чтобы изменить значение ***TOL*** для всего рабочего документа, выберите команду ***Математика ⇒Параметры… ⇒Переменные ⇒Допуск сходимости (TOL).***

Примеры использования функции ***root*** показаны на рис.7.

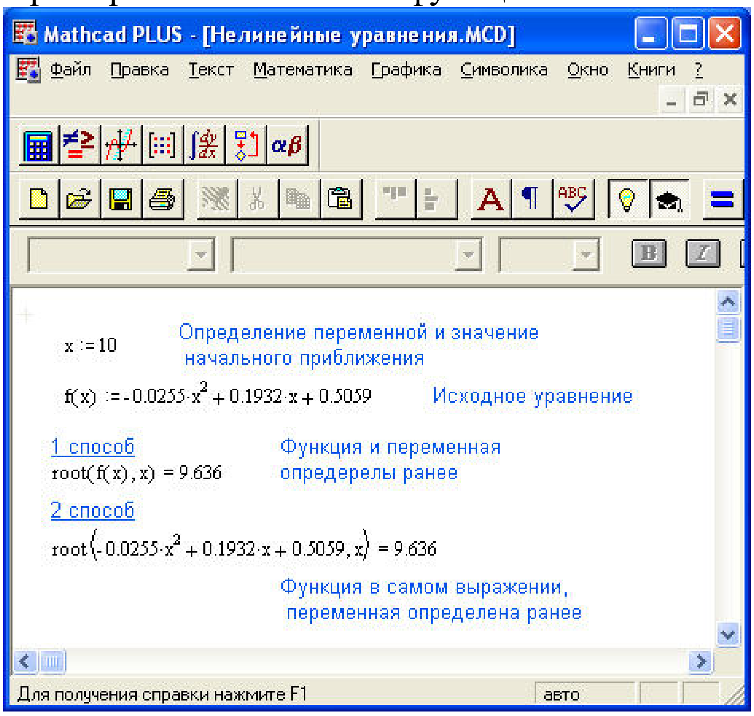


Рис.7. Рабочее поле для уточнения корня

Для нахождения корней выражения, имеющего вид

$$v_n x^n + ... + v_2 x^2 + v_1 x + v_0,$$

возможно использование функции ***polyroots***, нежели ***root***. В отличие от функции ***root***, функция ***polyroots*** не требует начального приближения и возвращает сразу все корни, как вещественные, так и комплексные. Но в приведенном задании этого не требуется.

Контрольные вопросы ПЗ3(ОПК-4):

1. Определение аппроксимации функции?
2. Определение интерполяции функции?
3. Условие минимума отклонения значений аппроксимирующей функции от экспериментальных точек?
4. Метод решения системы нормальных уравнений?
5. Методы нахождения корней системы нормальных уравнений?
6. Получение линейной и квадратичной аппроксимирующих функций?
7. Приведите структурную схему алгоритма метода наименьших квадратов для получения линейной и квадратичной аппроксимирующих функций.
8. Количественная оценка качества аппроксимации?
9. Какое приближение к таблично заданной функции считается лучшим?
10. Получение коэффициентов аппроксимации с использованием метода наименьших квадратов средствами пакета Excel и пакета MathCad?

Лабораторная работа №1. Уточнение корня нелинейного уравнения методом итерации средствами пакета Excel и средствами пакета MathCad.

Уточнение корня уравнения методом итерации

Для уточнения корня методом итерации с заданной точностью ε предполагается преобразование уравнения $F_2(x)=0$ к равносильному уравнению вида $x = \varphi(x)$. Если функция $\varphi(x)$ такая, что для всех *x* из интервала изоляции корней [k, m] выполняется неравенство $|\varphi'(x)| < 1$, то на данном отрезке может быть применен алгоритм простой итерации по формуле

$$x_{i+1} = \varphi(x_i).$$

При этом точным значением корня $x=x_т$ в геометрической интерпретации является абсцисса точки пересечения прямой $y_1=x$ и кривой $y_2=\varphi(x)$. Выбрав за начальное приближение к корню любое из интервала [n; k] значение $x=x_0$, необходимо произвести его итерационное уточнение последовательным вычислением значений $x_1=\varphi(x_0)$, $x_2=\varphi(x_1)$, … $x_n=\varphi(x_{n-1})$. Процесс таких вычислений оканчивается после выполнения проверяемого на каждом шаге итерации условия достижения точности $T=|\varphi(x_n)-x_n| \le \varepsilon$. За уточнённое при этом (с точностью ε) значение x* искомого корня принимается значение, равное x_n ($x^*= x_n$), отличающееся от точного $x_т$ на величину ε.

Для обеспечения условия сходимости функция должна иметь вид [1]:

$$\varphi(x) = \lambda \cdot F_2(x) + x,$$

где λ выбирается по правилу:

$$\begin{cases} -\frac{1}{r} < \lambda < 0, \text{ если } F_2'(x) > 0 \\ 0 < \lambda < \frac{1}{r}, \text{ если } F_2'(x) < 0 \end{cases}$$

При этом $r = \max(|F_2'(k)|, |F_2'(m)|)$.

Для отрезка [-2, -3] $F_2'>0$, r = max(|0,3462|, |0,2962|). Следовательно –(1/r) = -(1/0,3462) = -2,89. Поэтому λ выбирается из условия -2,89< λ<0. Выбираем λ = 2. Тогда последовательное приближение к корню вычисляется по формуле:

$$x_{n+1} = x_n - 2(-0.0255 \cdot x_n^2 + 0{,}1932 \cdot x_n + 0.5059),\ \ n = 0{,}1{,}2{,}...$$

В качестве начального приближения можно выбрать любую из точек границ отрезка [-3, -2], например x_0=-2.

Схема алгоритма для метода итерации представлена на рис.2.

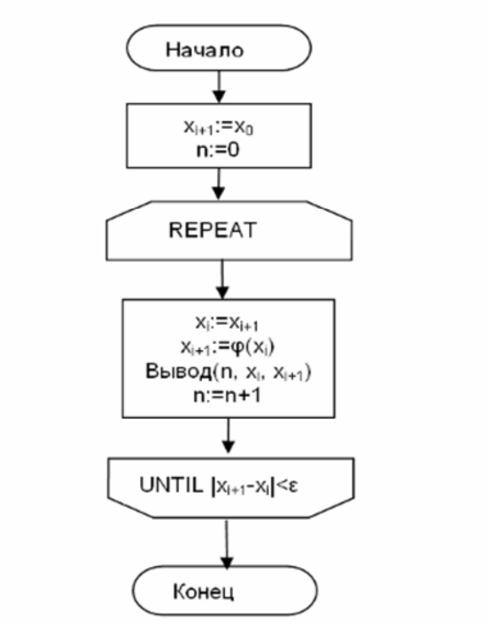


Рис.2. Схема алгоритма уточнения корня по методу итерации

Результаты вычислений представлены в таблице:

n	x_n	$\varphi(x_n)$.	T= $\lvert\varphi(x_n).- x_n\rvert$
0	-2,0000	-2,0350	0,0350
1	-2,0350	-2,0493	0,0143
2	-2,0493	-2,551	0,0058
3	-2,0551	-2,0574	0,0023
4	-2,0574	-2,0583	0,0009
5	-2,0583	-2,0587	0,0004
6	-2,0587	-2,0589	0,0002
7	-2,0589	Искомый корень	

Таким образом, уточненный корень на отрезке [-3, -2] x* = -2,0589. Подставляя его значение в исходную функцию, получаем:

$$F_2(-2{,}0589) = 2{,}43E\ -5.$$

Оценку погрешности метода можно выполнить по формуле скорости сходимости [2]:

$$|x^* - x_n| \le \frac{m}{1-q} \cdot q^n,$$

$$q = \max|\varphi'(x)|,$$

$$m = |x_0 - \varphi(x_0)| \qquad x \in [a, b]$$

m = |-2,0 - (-2,0350)| = 0,0350; n=7;

$$\varphi(x_n) = x_n - 2(-0.0255 \cdot x_n^2 + 0{,}1932 \cdot x_n + 0.5059),$$

$$\varphi'(x_n) = 1 - 2(-2 \cdot 0.0255 \cdot x_n + 0{,}1932)\ ,$$

$$q = \max\left|\ 1 - 2(-2 \cdot 0.0255 \cdot (-2{,}0) + 0{,}1932)\ \right| = 0{,}4096$$

Тогда погрешность

$$|x^* - x_n| \le \frac{0{,}0350}{1 - 0{,}4096} \cdot 0{,}4096^7 = 1{,}15E-4$$

Пример исходного текста программы для уточнения корней по методу итерации представлен в приложении.

Уточнение корня уравнения методом итерации средствами пакета □□□l

Способ 1. С появлением математических пакетов и электронных таблиц стало возможным вычислять таблицы значений функции с любым шагом и строить графики с высокой точностью. Это позволяет уточнять очередной знак в приближенном значении корня при помощи следующего алгоритма:

1) Если функция f(x) на концах отрезка [a, b] принимает значения разных знаков, то разделить отрезок на 10 равных частей и найти ту часть, которая содержит корень (таким способом возможно уменьшить длину отрезка, содержащего корень, в 10 раз);

2) Повторить действия предыдущего пункта для полученного отрезка. Процесс продолжать до тех пор, пока длина отрезка не станет меньше заданной погрешности.

Технически этот алгоритм реализуется по методике построения графиков функций в Excel:

1. Ввести значения аргумента *x*. Для этого в ячейке A2 записать первое значение аргумента, в ячейке A3 — последующее значение аргумента с заданным шагом. После этого выделить диапазон A2:A3 и с помощью маркера заполнения присвоить значения остальным ячейкам до необходимого значения аргумента;

2. В ячейку B2 ввести формулу "=F_2(x)" и скопировать функцию B2 с помощью маркера заполнения в остальные ячейки по количеству значений аргументов;

3. Построить с помощью мастера диаграмм график функции (тип диаграммы "График с маркерами").

Способ 2. При наличии итерационной формулы (т.е. обеспечении условия сходимости) и отрезка отделения корня значение корня можно уточнить с заданной точностью, построив специальным образом таблицу расчетов. Ее структура и содержание данных должны обеспечить следующие условия:

1. Каждая строка таблицы представлять собой полностью данные одного шага итерации;

2. Для простоты визуального контроля достижения требуемой точности желательно предусмотреть соответствующие средства на каждом шаге итерации;

3. Наращивание количества шагов итерации должно быть возможно увеличением числа строк таблицы, и, кроме того, это наращивание для простоты должно осуществляться выполнением элементарной операции копирования строк.

Для рассматриваемой задачи:

- отделенный отрезок – [-3, -2], начало процесса от x=-2;
- итерационная формула

$\varphi(x_n) = x_n - 2(-0.0255 \cdot x_n^2 + 0{,}1932 \cdot x_n + 0.5059),$

- точность рассчитывается по формуле - $T=|\varphi(x_n)-x_n|\leq\varepsilon;$
- точность 10^{-3}.

Таблица формируется в следующем порядке (см. рис.1):

1. Вводятся надписи в шапке таблицы: n – порядковый номер итерации (A1); значение переменной (B1); значение итерирующей функции при этой переменной (C1); значение точности на шаге итерации (D1); информационное сообщение на данном шаге (E1).

	A	B	C	D	E
1	n	x_n	$\varphi(x_n)$	$\|\varphi(x_n)-x_n\|$	Сообщение
2	0	-2	=B2-2*(-0,0255*B2*B2+0,1932*B2+0,5059)	=ABS(C2-B2)	=ЕСЛИ(D2>0,001;"Считать далее";"Корень найден = "&C2)
3	1	=C2			
4					

Рис.1. Формулы для уточнения корня методом итерации

2. Заполняется первая строка таблицы (A2:E2) по правилам получения соответствующих данных Первая строка, как правило, имеет начальные значения переменных (например, x_n=-2 в ячейке B2) и этим отличается от остальных строк. Поэтому ее копировать нельзя, но на ее основе строится вторая строка, которая для копирования пригодна;

3. Во второй строке таблицы порядковый номер может быть проставлен константой (A3) как на рисунке или вычислен по формуле "=A2+1". В последнем случае она может быть в дальнейшем скопирована в составе всей строки. Значение новой переменной (ячейка B3) в соответствии с процедурой итерации должно быть равно значению итерирующей функции на предыдущем шаге, поэтому вычисляется по формуле "=C2". Остальные формулы второй строки копируются из выше расположенных ячеек;

4. Вторая строка выделяется как диапазон A3:E3 и маркером автозаполнения копируется вниз до появления сообщения о достижении требуемой точности. Если точность не достигнута, то копирование продолжается. Увидеть все введенные формулы возможно при выполнении команд ***Сервис-Зависимости формул-Режим проверки формул***

Результаты расчета по введенным формулам представлены в таблице на рис.2. Легко заметить, что точность на шаге №4 выше, чем 0,001.

	A	B	C	D	E
1	n	x_n	$\varphi(x_n)$	$\|\varphi(x_n)-x_n\|$	Сообщение
2	0	-2	-2,0350	0,0350	Считать далее
3	1	-2,0350	-2,0493	0,0143	Считать далее
4	2	-2,0493	-2,0551	0,0058	Считать далее
5	3	-2,0551	-2,0574	0,0023	Считать далее
6	4	-2,0574	-2,0583	0,0009	Корень найден = -2,05834195922734
7					

Рис.2. Результаты уточнения корня по методу итерации

Способ 3. Он предполагает использование средств электронной таблицы, специально рассчитанных на быстрый и простой поиск корней уравнения непосредственно в визуальной среде. Метод, применяемый в Excel для решения таких уравнений – модифицированный конечными разностями метод Ньютона, который рассмотрен в следующем подразделе. Метод позволяет не сильно заботиться о начальном приближении, как этого требуют другие численные методы решения уравнений (метод хорд, дихотомии и др.). Единственно, что следует учесть - это то, что будет найдено решение ближайшее к выбранному начальному приближению.

Однако пользователь не видит, каким методом осуществляется поиск корней. Поэтому для полноты картины эта технология рассмотрена в этом разделе:

1. Занесите в ячейку А1 значение 0;
2. Занесите в ячейку В1 левую часть уравнения, используя в качестве независимой переменной ссылку на ячейку А1. Соответствующая таблица для рассматриваемого уравнения показана на рис.3.

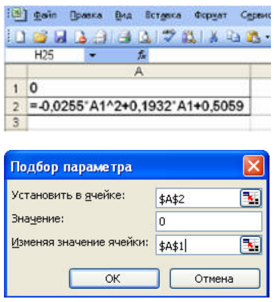


Рис.3. Поиск корня по технологии ***Подбор параметра***

3. Выполните команду ***Сервис-Подбор параметра***. В поле ***Установить в ячейке*** укажите В1, в поле ***Значение*** – задайте "0", в поле ***Изменяя значение ячейки*** укажите А1. Должно получиться окно представленное на рис.4.
4. После нажатия ***Ок*** результаты подбора отобразятся в окне ***Результат подбора параметра***. В ячейке А1 будет записано значение переменной, при которой выражение в ячейке А2 приобрело указанное значение 0 (т.е. обратилось в равенство). Это и есть по определению корень уравнения (см. значения на рис.4). В ячейке А2 значение после подбора несколько отличается от нуля, но это результат приближенного решения, и значение это мало. В соответствии с особенностями алгоритма этот корень ближайший к начальному значению нуль, который был введен в ячейку А1. После нажатия ***Ок*** еще раз – участвующие в подборе ячейки будут сохранены.

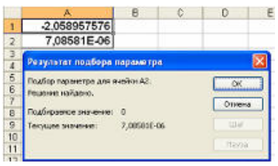


Рис.4. Результаты подбора параметра

Если в ячейке А1 при запуске подбора указать значение, например, 7, то результат будет иным. Объяснить это попытайтесь самостоятельно.

Численные методы решения уравнений хороши тем, что позволяют получить приближенное решение с задаваемой точностью. Для этого надо выполнить команду ***Сервис-Параметры-Вычисления*** и в соответствующих позициях установить значения относительной погрешности и количества итераций.

Контрольные вопросы ЛР1(ОПК-4):

6. В чём заключается задача численного интегрирования?
7. Как группируются численные методы интегрирования?
8. Уточнение корня уравнения методом итерации средствами пакета Excel первый способ?
9. Уточнение корня уравнения методом итерации средствами пакета Excel второй способ?
10. Вычисление интеграла при помощи пакета Excel?

11. Вычисление интеграла при помощи пакета MathCad?
12. Правило Рунге оценивания погрешности вычисления интеграла?
13. Сравнение методов численного интегрирования?

Лабораторная работа №2. Оценка погрешностей вычисления корней при решении задач на ПК, связанных с решением нелинейных уравнений методами половинного деления, итерации и касательных.

1. Для выполнения работы на плоскости X,Y задаётся последовательность экспериментально полученных точек, именуемых дискретной функцией Y(x). Она представляется таблицей:

i	0	1	2	3	4	5
X_i	-6	-4	-2	0	2	4
$Y(x)_i$	-1,060	-1,406	-0,386	1,000	1,386	0,406

Предлагается считать, что набор представленных экспериментальных данных получен со значительной погрешностью. Поэтому отвергается возможность интерполирования заданной функции Y(x), так как основным условием интерполяции является равенство значений интерполируемой Y(x) и интерполирующей φ(x) функций в узлах интерполяции Xi , т.е. Y(Xi)= φ(Xi). С учётом изложенного заданием №1 предписывается:

- выполнить регрессионный анализ заданной функции Y(x). Из множества его видов выбрать степенной вид регрессии и методом наименьших квадратов получить линейную $F_1(x) = a_0 + a_1 x$ и квадратичную $F_2(x) = a_0 + a_1 x + a_2 x^2$ аппроксимирующие функции;
- составить и решить системы линейных алгебраических (нормальных) уравнений для определения параметров a_i функций $F_1(x)$ и $F_2(x)$;
- вычислить значения полученных аппроксимирующих функций в узлах аппроксимации x_i ;
- построить графики дискретно заданной функции (в виде множества точек) и графики найденных функций линейной и квадратичной аппроксимации;
- оценить качество выполненной аппроксимации.

О приближениях функций по методу наименьших квадратов

С целью повышения достоверности математического описания результатов эксперимента этими функциями необходимо определить их параметры a_i таким образом, чтобы величина всех отклонений графиков функций $F_m(x)$ от экспериментальных точек y_j в узлах x_i была минимально возможной и обладала при этом (согласно требованию задания) наименьшей среднеквадратичной погрешностью, то есть отвечала бы условию:

$$U = \sum_{i=1}^{N} [y_i - f_m(x_i)]^2 = \min \qquad (3.1)$$

Математическое выражение условия (3.1) является основой для определения требуемых параметров a_i аппроксимирующих функций $F_1(x)$ и $F_2(x)$. Для этого следует:

1. Подставить в (3.1) выражение функции $F_1(x)$ и получить

$$U_1 = \sum_{i=1}^{N} [y_i - a_0 - a_1 x_i]^2;$$

2. Получить выражения частных производных от U_1 по искомым параметрам a_0 и a_1 в виде:

$$\frac{\partial U_1}{\partial a_0} = \sum_{i=1}^{N} [y_i - a_0 - a_1 x_i],$$

$$\frac{\partial U_1}{\partial a_1} = \sum_{i-1}^{N} [y_i - a_0 - a_1 x_i] \cdot x_i;$$

3. Приравнять выражения этих производных нулю и после простейших преобразований получить систему двух (с неизвестными a_0 и a_1) линейных алгебраических (нормальных) уравнений вида:

$$\left\{ \begin{aligned} & a_0 \sum_{i=1}^{N} x_i^0 + a_1 \sum_{i=1}^{N} x_i = \sum_{i=1}^{N} y_i \\ & a_0 \sum_{i=1}^{N} x_i + a_1 \sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} x_i y_i \end{aligned} \right. \qquad (3.2)$$

4. Подставить снова в (3.1) выражение теперь второй функции $F_2(x)$ и получить:

$$U_2 = \sum_{i=1}^{N} [y_i - a_0 - a_1 x_i - a_2 x_i^2]^2;$$

5. Получить выражения частных производных от U_2 по трём искомым параметрам a_0 , a_1 и a_2 , приравнять их нулю и после простейших преобразований получить систему трёх нормальных уравнений вида:

$$\left\{ \begin{aligned} & a_0 \sum_{i=1}^{N} x_i^0 + a_1 \sum_{i=1}^{N} x_i + a_2 \sum_{i=1}^{N} x_i^2 = \sum_{i=1}^{N} y_i \\ & a_0 \sum_{i=1}^{N} x_1 + a_1 \sum_{i=1}^{N} x_i^2 + a_2 \sum_{i=1}^{N} x_i^3 = \sum_{i=1}^{N} x_i y_i \\ & a_0 \sum_{i=1}^{N} x_i^2 + a_1 \sum_{i=1}^{N} x_i^3 + a_2 \sum x_i^4 = \sum_{i=1}^{N} x_i^2 y_i \end{aligned} \right. \qquad (3.3)$$

6. Решить полученные системы (3.2) и (3.3) нормальных уравнений, предварительно рассчитав и получив числовые значения их (находящихся под знаками суммы) коэффициентов и свободных (расположенных после знаков равенства) членов.

Коэффициенты и свободные члены нормальных уравнений

Для упрощения процесса решения полученных систем (3.2) и (3.3) нормальных уравнений целесообразно предварительно определить (рассчитать и получить) числовые значения их коэффициентов и свободных членов. Все промежуточные и конечные результаты их вычислений для заданного варианта программно получены (см. приложение) и наглядно представлены в таблице 3.1 (матрице Грамма):

Таблица 3.1

i	x^0	x^1	x^2	x^3	x^4	y	y*x	$y*x^2$
0	1	-6	36	-216	1296	-1,060	6,360	-38,160
1	1	-4	16	-64	256	-1,406	5,624	-22,496
2	1	-2	4	-8	16	-0,386	0,772	-1,544
3	1	0	0	0	0	1,000	0,000	0,000
4	1	2	4	8	16	1,386	2,772	5,544
5	1	4	16	64	256	0,406	1,624	6,496
Σ	6	-6	76	-216	1840	-0,060	17,152	-50,160

Решение систем нормальных уравнений

Приведенные в последней строке таблицы 3.1 числовые значения коэффициентов и свободных членов систем нормальных уравнений (3.2) и (3.3) позволяют представить их в пригодном для решения по заданным условиям в виде:

$$\begin{cases} 6 \cdot a_0 - 6 \cdot a_1 = -0{,}060 \\ -6 \cdot a_0 + 76 \cdot a_1 = 17{,}152 \end{cases} \qquad (3.4)$$

$$\begin{cases} 6 \cdot a_0 - 6 \cdot a_1 + 76 \cdot a_2 = -0{,}060 \\ -6 \cdot a_0 + 76 \cdot a_1 - 216 \cdot a_2 = 17{,}152 \\ 76 \cdot a_0 - 216 \cdot a_1 + 1840 \cdot a_2 = -50{,}160 \end{cases} \qquad (3.5)$$

Для решения таких систем можно выбрать любой из множества разработанных численных методов (например, метод Гаусса, метод Гаусса-Жордана, метод прогонки, метод квадратных корней, метод итерации, метод Крамера и другие тому подобные). Наиболее подходящим для решения системы (3.4) является матричный метод Крамера, а системы (3.5) – метод Гаусса.

Метод Крамера позволяет находить решение системы линейных уравнений с помощью определителей. Если систему уравнений (3.4) представить в матричной форме:

$$\begin{vmatrix} 6 & -6 \\ -6 & 76 \end{vmatrix} * \begin{vmatrix} a_0 \\ a_1 \end{vmatrix} = \begin{vmatrix} -0.060 \\ 17.152 \end{vmatrix},$$

то её определителями будут:

$$\begin{vmatrix} 6 & -6 \\ -6 & 76 \end{vmatrix} = 6 * 76 - (-6)*(-6) = 456 + 36 = 420 = \Delta,$$

$$\begin{vmatrix} -0.060 & -6 \\ 17.152 & 76 \end{vmatrix} = (-0.060) * 76 - 17.152 * (-6) = 98.352 = \Delta a_0,$$

$$\begin{vmatrix} 6 & -0.060 \\ -6 & 17.152 \end{vmatrix} = 6 * 17.152 - (-6) * (-0.060) = 102.552 = \Delta a_1.$$

Используя далее известные формулы Крамера, получаем значения корней системы уравнений (3.4). Ими являются искомые параметры аппроксимирующей функции $F_1(x)$:

$$a_0 = \Delta\, a_0 / \Delta = 98,352 / 420 = 0,234;$$
$$a_1 = \Delta\, a_1 / \Delta = 102,552 / 420 = 0,244.$$

Методом Крамера может быть решена и система уравнений (3.5). Однако из-за большого объёма вычислений (существенно возрастающего с увеличением числа уравнений в системе) применение этого метода не рекомендуется. Поэтому лучше выбрать свободный от этого недостатка численный метод Гаусса.

Метод Гаусса являются наиболее экономичными для программного решения систем с неограниченным числом линейных уравнений. В основе метода Гаусса лежит идея последовательного исключения неизвестных. Данная идея реализуется вычислительной схемой единственного деления, состоящей из прямого и обратного ходов. Процесс решения системы нормальных уравнений сводится к её преобразованию в эквивалентную систему уравнений в виде треугольной матрицы (прямой ход) с последующим нахождением решения (обратный ход) в виде конкретных значений всех неизвестных этой системы.

Прямой ход выполняется приведением коэффициентов при одноименных переменных a_i к одинаковому значению путем деления левой и правой части соответствующих уравнений на коэффициент при этой переменной с последующим сложение этих уравнений.

Программное (см. приложение) решение системы уравнений (3.5) методом Гаусса выдало её сначала в треугольной форме

$$\begin{cases} 1\cdot a_0 - 1\cdot a_1 + 12{,}667\cdot a_2 = -0{,}010 \\ \qquad\quad 1\cdot a_1 - \quad 2\cdot a_2 = 0{,}244 \\ \qquad\qquad\quad 597{,}333\cdot a_2 = -15{,}216 \end{cases} \qquad (3.6)$$

а затем, решив её, числовые значения искомых корней:

$$a_0 = 0,506; \quad a_1 = 0,193; \quad a_2 = -0,025.$$

Контрольные вопросы ЛР2(ОПК-4):

1. Как производится этап отделения корней?
2. На какой теореме основывается аналитическое отделение корней?
3. При каких условиях корень будет единственным на отрезке отделения корней?
4. В чём заключается этап уточнения корней?
5. В чём суть метода половинного деления?
6. Что является условием окончания поиска корня?
7. В чём суть метода простой итерации уточнения корня?
8. Какие условия должны быть выполнены, чтобы процесс сходился к корню?
9. Что является условием окончания поиска корня в методе простой итерации?
10. Сравните методы уточнения корней.
11. Приведите структурную схему алгоритма заданного метода, уточняющего корень нелинейного уравнения.
12. Как выполняется оценка погрешностей вычисления корней

Лабораторная работа №3. Аппроксимация функции методом наименьших квадратов в соответствии с заданием. Составление алгоритма и программы решения. Решение задания средствами математических пакетов.

Функции и графики функций линейной и квадратичной аппроксимации

Полученные в предыдущем разделе числовые значения параметров a_i позволяют представить в окончательном виде функции линейной $F_1(x)$ и квадратичной (гиперболической) $F_2(x)$ аппроксимации. Ими являются:

$$F_1(x)=a_0+a_1x = 0,234 + 0,244x, \qquad (3.7)$$

$$F_2(x)=a_0+a_1x+a_2x^2 = 0,506 + 0,193x - 0,025x^2. \qquad (3.8)$$

Значения этих функций в узлах аппроксимации $F_1(x_i)$ и $F_2(x_i)$ (совместно с заданными Y_i приведены в таблице 3.2.

Таблица 3.2

i	0	1	2	3	4	5
x_i	–6	–4	–2	0	2	4
y_i	–1,060	–1,406	–0,386	1,000	1,386	0,406
$F_1(x_i)$	–1,231	–0,743	–0,254	0,234	0,723	1,211
$F_2(x_i)$	–1,571	–0,675	0,018	0,506	0,790	0,871

Из данной таблицы легко определить значения отклонений $\delta = F_m(x_i) - y_i$, которые характеризуют степень приближения полученных аппроксимирующих функций к таблично заданной. Наглядность такого приближения обеспечивается графическим представлением этих функций в прямоугольной системе координат X,Y. Построенные с помощью табличного процессора Excel графики этих функций заданного варианта показаны на рис.3.1.:

Количественная оценка качества аппроксимации рассчитывается по формуле $\rho = \sqrt{\frac{1}{n+1}\sum_{i=0}^{N}(F_m(x_i) - y(x_i))^2}$. Программно выполненные расчеты по этой формуле (см. программу в приложении) выдали результаты ρ_1=0.60018 для линейной и ρ_2 =0.54370 для квадратичной аппроксимации.

Так как $\rho_2 < \rho_1$, то следует считать, что несколько лучшим приближением к таблично заданной функции обладает полученная функция квадратичной аппроксимации вида:

$F_2(x) = 0.506 + 0.193x - 0.025x^2$.

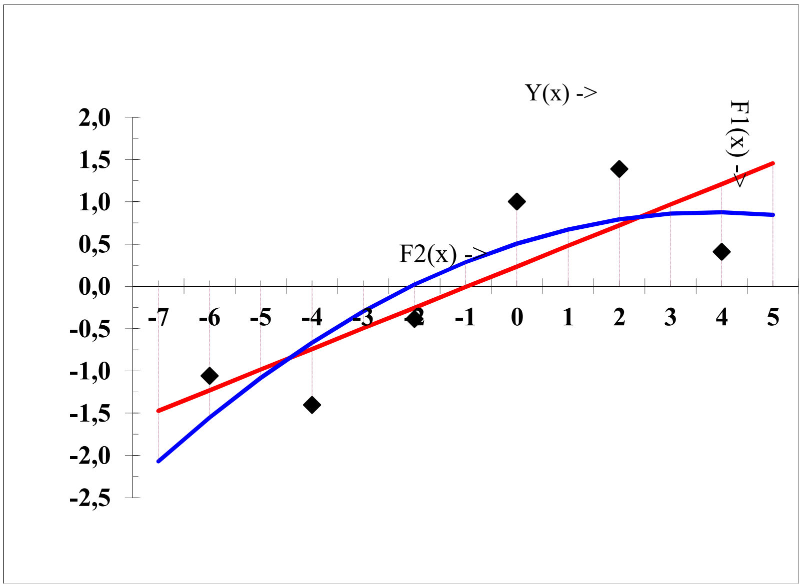


Рис.3.1 Графики функций линейной и квадратичной аппроксимации
на фоне точек таблично заданной функции

Получение коэффициентов аппроксимации с использованием метода наименьших квадратов средствами пакета Excel

Для заданного набора пар значений независимой переменной x_i и функции yi определить наилучшее линейное приближение в виде прямой с уравнением y=kx+b. Эту задачу решает функция ***ЛИНЕЙН***. Поскольку возвращается массив значений (k и b), функция должна задаваться в виде формулы массива.

Синтаксис функции для решения этой задачи следующий:

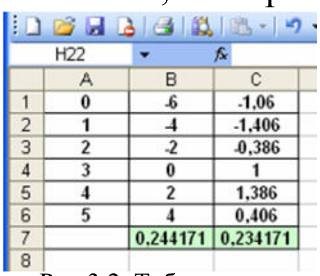
ЛИНЕЙН(известные_знач_y;известные_знач_x;конст;статист)

Известные_знач_y — множество значений y, которые уже известны для соотношения y = kx + b. Для табличного задания функции (рис.8.2) это диапазон C2:C7;

Известные_знач_x — необязательное множество значений x, которые уже известны для соотношения y = kx + b. На том же рисунке это диапазон B2:B7. Количество переменных *x* и *y* должно быть одинаково;

Конст — логическое значение, которое указывает, требуется ли, чтобы константа *b* была равна 0. Если ***конст*** имеет значение ИСТИНА или опущено, то b вычисляется обычным образом. Если аргумент ***конст*** имеет значение ЛОЖЬ, то b полагается равным 0 и значения k подбираются так, чтобы выполнялось соотношение y = kx;

Статистика — логическое значение, которое указывает, требуется ли вернуть дополнительную статистику по регрессии. Если аргумент имеет значение ЛОЖЬ или опущен, то функция ЛИНЕЙН возвращает только коэффициенты k и постоянную b.


H22

	A	B	C
1	0	-6	-1,06
2	1	-4	-1,406
3	2	-2	-0,386
4	3	0	1
5	4	2	1,386
6	5	4	0,406
7		0,244171	0,234171
8			

Рис.3.2. Табличное задание функции

Для формирования формулы необходимо для ячейки B7 выполнить ***Вставка функции-Статистические-ЛИНЕЙН.*** В поле ***Известные_значения_y*** и ***Известные_значения_x*** ввести соответствующие диапазоны. Остальные поля можно опустить. Получится окно ***Аргументы функции*** такое, как показано на рис.3.3. При вычислении в ячейке B7 получится значение коэффициента k=0,24417, что соответствует предыдущим расчетам.

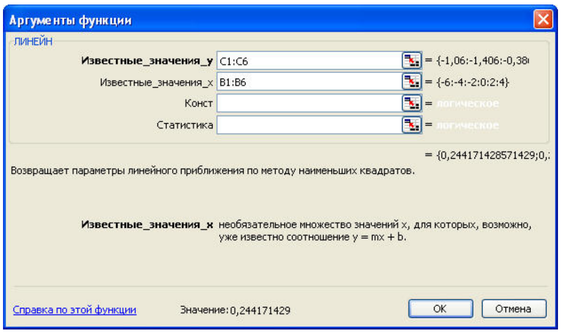


Рис.3.3. Аргументы функции ЛИНЕЙН

Если формула не будет введена как формула массива, получится только один коэффициент. Поэтому после получения коэффициента k выделите диапазон A7:C7, начиная с ячейки, содержащей формулу. Нажмите клавишу ***F2,*** а затем — клавиши ***CTRL+S□IFT+Enter***. Получится второе число, соответствующее свободному члену b уравнения вида Y=Kx+b (см. ячейку C7 на рис.3.2).

Аналогичный результат получается, если сформировать функцию вида ***=ИНДЕКС(ЛИНЕЙН(C1:C6;B1:B6);1)*** для коэффициента k и аналогичную с аргументом 2 для получения коэффициента b линейной функции. Функция ЛИНЕЙН возвращает коэффициенты уравнения прямой в виде массива из двух элементов. С помощью функции ИНДЕКС выбирается нужный элемент.

Однако самым эффективным способом получения коэффициентов аппроксимации является механизм расчета трендов (. Тренд от англ. Trend — тенденция). Линии тренда представляют собой геометрическое отображение средних значений анализируемых показателей, полученное с помощью какой-либо математической функции (линейной, полиномиальной и т.д.). Выбор функции для построения линии тренда обычно определяется характером изменения данных во времени.

Для получения уравнения тренда необходимо по заданным точкам построить график функции. Установить курсор на линию кривой и правой кнопкой выбрать ***Добавить линию тренда***. В окне ***Линия тренда*** (рис.3.4) на закладке ***Тип*** выбрать тип аппроксимирующей кривой, а на закладке ***Параметры*** – установить флажок в окне ***Показать уравнение на диаграмме.*** На графике добавится линия тренда и уравнение аппроксимирующей кривой с искомыми коэффициентами.

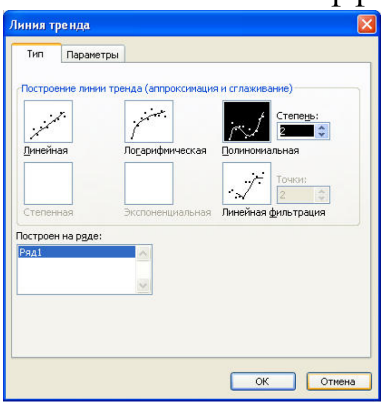


Рис.3.4. Окно линии тренда

Получение коэффициентов аппроксимации с использованием пакета MathCad

Для решения системы уравнений в MathCad используется функция поиска точного решения ***Find(x1,x2,x3)***. Однако в этом случае необходимо задать для всех неизвестных начальные приближения. Это удобно далеко не всегда.

Исходная система полученных линейных уравнений (3.5) имеет вид

$$\begin{cases} 6 \cdot a_0 - 6 \cdot a_1 + 76 \cdot a_2 = -0{,}060 \\ -6 \cdot a_0 + 76 \cdot a_1 - 216 \cdot a_2 = 17{,}152 \\ 76 \cdot a_0 - 216 \cdot a_1 + 1840 \cdot a_2 = -50{,}160 \end{cases}$$

В матричной форме элементы этой системы должна быть представлена в виде:

$$A = \begin{bmatrix} 6 & -6 & 76 \\ -6 & 76 & -216 \\ 76 & -216 & 1840 \end{bmatrix}, \qquad b = \begin{bmatrix} -0{,}060 \\ 17{,}152 \\ -50{,}160 \end{bmatrix}.$$

В соответствии с правилом умножения матриц рассмотренная система линейных уравнений может быть записана в матричном виде

$$Ax = b, \tag{3.9}$$

где: $a = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}.$

Матрица A, столбцами которой являются коэффициенты при соответствующих неизвестных, а строками - коэффициенты при неизвестных в соответствующем уравнении, называется матрицей системы; матрица-столбец b, элементами которой являются правые части уравнений системы, называется правой частью системы (матрицей свободных членов). Матрица-столбец *a*, элементы которой - искомые неизвестные, называется решением системы (неизвестных переменных).

Если матрица A - неособенная, то есть ее определитель det A (или |A|) не равен нулю, система (3.5), или эквивалентное ей матричное уравнение (3.9), имеет единственное решение. Ранг неособенной матрицы равен ее порядку.

В самом деле, при условии det A ≠ 0 существует обратная матрица A^{-1}. Умножая обе части уравнения (3.9) на матрицу A^{-1} получим:

$$A^{-1} \cdot A \cdot x = A^{-1} \cdot b \tag{3.10}$$

$$x = A^{-1} \cdot b$$

Формула (3.10) дает решение уравнения (3.9) и оно единственно (рис.3.5).

Определитель матрицы |A| = 6*76*1824 + (-6)*(-216)*76 + (-6)*-216)*76 - 76*76*76 + (-6)*(-6)*1840 – (-216)*(-216)*6. Обратная матрица – это та, умножение на которую исходной матрицы дает единицу.

На рис.3.5. матрица AA является обратной по отношению к матрице A. Для этого выполнены операции ***Символика-Матричные_операции-Обратить***. Проще это можно сделать, вычислив A^{-1}.

Стандартным средством решения линейных уравнений является встроенная функция ***lsolve(A,b)***, которая решает систему на основе LU-разложения матрицы (с частичным выбором главного элемента). Аргументами функции являются квадратная невырожденная матрица A и вектор правых частей b (столбцевая матрица). Возвращается вектор решения *a* такой, что $Aa = b$. Форматы выражений показаны на рис.3.6. Обратите внимание, что в выражении используется малая латинская буква l. Результаты решения на рис.3.5 и рис.3.6 совпадают с ранее полученными. Проверка на рис.3.6 показала совпадение результата умножения с матрицей свободных членов, что также подтверждает правильность найденных коэффициентов.

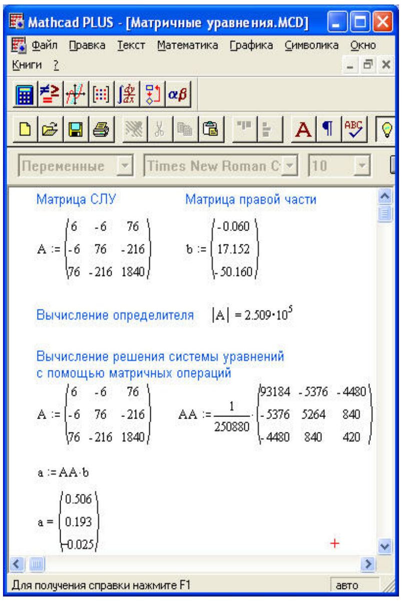


Рис.3.5. Решение уравнения при помощи аппарата матриц

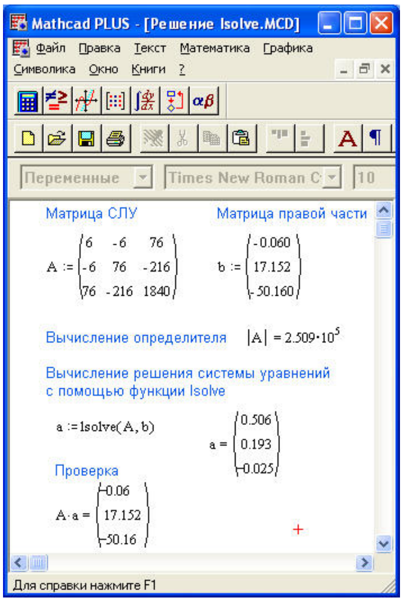


Рис.3.6. Использование функции ***lsolve***

Контрольные вопросы ЛР3(ОПК-4):

1. Что такое аппроксимация функции?
2. Определение интерполяции функции?
3. Условие минимума отклонения значений аппроксимирующей функции от экспериментальных точек?
4. Метод решения системы нормальных уравнений?
5. Методы нахождения корней системы нормальных уравнений?
6. Получение линейной и квадратичной аппроксимирующих функций?
7. Приведите структурную схему алгоритма метода наименьших квадратов для получения линейной и квадратичной аппроксимирующих функций.
8. Количественная оценка качества аппроксимации?
9. Какое приближение к таблично заданной функции считается лучшим?
10. Получение коэффициентов аппроксимации с использованием метода наименьших квадратов средствами пакета Excel и пакета MathCad?

Практическое занятие №4. Методы численного интегрирования решения задач. Составление алгоритма и программы решения. Решение задания средствами математических пакетов.

1 Условие задания

Используя численные методы средних прямоугольников, трапеций и квадратичных парабол (Симпсона), вычислить значение определённого интеграла

$$I = \int_{X_1^*}^{X_2^*} F_2(x)\,dx,$$

где - $F_2(x)$ – функция квадратичной аппроксимации (найденная в задании №1),

- x_1^* и x_2^* - корни уравнения $F_2(x)=0$ (найденные в задании №2).

Результаты вычисления интеграла каждым методом представить для случаев разбиения интервала его интегрирования $[x_1^*;\ x_2^*]$ на n=10 и n=20 отрезков одинаковой длины h. Представить также результаты оценки погрешности вычислений интеграла перечисленными методами интегрирования.

2. Методика вычисления исходного интеграла

Исходным является определённый интеграл вида:

$$I = \int_{X_1^*}^{X_2^*} (-0.025x^2 + 0.193x + 0.506)\,dx,$$

где $x_1^*=-2{,}0589$ и $x_2^*=9{,}6355$-нижний и верхний пределы интегрирования.

Известно, что в основе поиска его конкретного значения любым методом численного интегрирования лежит обычно приближённое вычисление площади плоской геометрической фигуры, ограниченной графиком подынтегральной функции $F_2(x)$ в пределах интервала $[x_1^*; x_2^*]$ изменения аргумента x.

Чаще всего, получаемая таким образом фигура имеет неопределённую криволинейную форму, для вычисления площади которой не существует формул. Поэтому и прибегают к различным методам изменения этой формы, среди которых наиболее популярными считаются методы прямоугольников, трапеций и квадратичных парабол. Получение геометрических фигур, соответствующих названиям этих методов, основано на замене любой подынтегральной функции интерполяционным многочленом

$$\varphi(x)=\beta_0 x_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

соответственно нулевой, первой и второй степени.

Для повышения точности вычисления интеграла любым из перечисленных методов целесообразно:

- разбить интервал интегрирования $[x_1^*;\ x_2^*]$ на n одинаковых отрезков длиной $h=(x_2^* - x_1^*]/n$ и получить в нём серию конкретных значений аргумента x вида $x_0=x_1^*$, $x_1=x_0+h$, $x_2=x_1+h$, $x_3=x_2+h$, …, $x_i = x_{i-1}+h$, …, $x_n=x_2^*$;

- разбить абсциссами точек x_i этой серии общую криволинейную фигуру на n подфигур и в каждую из них вписать прямоугольник, трапецию или криволинейную трапецию (в две рядом расположенные подфигуры);

- вычислить площади каждой вновь полученной геометрической фигуры и, просуммировав их, получить приближённый результат вычисления интеграла.

Контрольные вопросы ПЗ4(ОПК-4):

1. В чём заключается задача численного интегрирования?
2. Как группируются численные методы интегрирования?
3. Структурные схемы алгоритмов заданных методов прямоугольников?
4. Структурная схема алгоритма метода трапеций?
5. Структурная схема алгоритма метода Симпсона?
6. Вычисление интеграла при помощи пакета Excel?
7. Вычисление интеграла при помощи пакета MathCad?

Практическое занятие №5. Численное интегрирование. Структурные схемы алгоритмов, программирование и решение задач на ПК, связанных с вычислением определённых интегралов методами прямоугольников, трапеций и Симпсона.

1. Условие задания

Применяя численные методы дихотомии и золотого сечения одномерной оптимизации, определить с точностью до $\varepsilon = 10^{-3}$ точку экстремума функции квадратичной аппроксимации F_2 (x), из задания №1. При выполнении задания предусмотреть:

1. проверку условия унимодальности функции $F_2(x)$;
2. обоснованный выбор начального отрезка оптимизации;
3. обоснование условий окончания поиска экстремума $F_2(x)$ в используемых методах одномерной оптимизации.

2. Характеристика исходной функции

Исходной является функция квадратичной параболы вида:

$$F_2(x) = -0.025 X^2 + 0.193 X + 0.506$$

с производными $F_2'(x) = -0.050 X + 0.193$ и $F_2''(x) = -0.050.$

Так как | $F_2''(x)$ | = 0,0510> 0 , то исходная функция является *унимодальной*, то есть содержащей в пределах изменения аргумента ***x*** от “минус” до “плюс” бесконечности только один экстремум. Из анализа усечённого графика исходной функции следует, что её экстремумом является максимум. В результате программной табуляции этой функции установлено, что он расположен на интервале оптимизации [Ni=3,0; Ki=5,0], где Ni и Ki - начало и конец интервала предстоящего поиска экстремума.

Для поиска экстремумов унимодальных функций разработано множество разнообразных методов. Задача поиска в каждом из них сводится к локализации и уточнению значений x и $F_2(x)$ в точке экстремума. Одни методы рассчитаны на поиск максимумов функций, другие – на поиск их минимумов. Но поскольку максимуму функции $F_2(x)$ соответствует минимум $-F_2(x)$, то, сменив знак у $F_2(x)$, алгоритмами и программами поиска максимума (минимума) любым методом можно пользоваться и для поиска минимума (максимума) любой унимодальной функции.

Методы одномерной оптимизации используют следующее свойство непрерывной функции: если точки g и h ($g<h$) расположены на отрезке $[a, b]$ и $f(g) < f(h)$, то на отрезке $[a, h]$ есть хотя бы один минимум функции, если $f(g) > f(h)$, то на отрезке [g, b] есть хотя бы один минимум. В силу унимодальности функции этот минимум будет единственным. Тем самым уменьшается отрезок поиска минимума. Повторяя указанный процесс, получают последовательность уменьшающихся отрезков. Поиск минимума можно свести к определению отрезка, длина которого станет меньше заданной точности. Это будет окончанием поиска $|b_k - a_k| < \varepsilon$.

3. Поиск экстремума методом дихотомии

Метод дихотомии (метод деления интервала поиска пополам) может быть реализован алгоритмом поиска экстремума (например, максимума) вида:

1. Задаются конкретные значения границ N_i и K_i интервала поиска максимума, значение ε погрешности его вычисления и значения коэффициентов a_0, a_1, a_2 функции $F_2(x)$;
2. Составляется подпрограмма вычисления $F_2(x)$ для задаваемых ей значений аргумента x.
3. Делится пополам интервал поиска $[N_i; K_i]$ и относительно получаемого при этом среднего значения аргумента $x=(N_i+K_i)/2$ вычисляются две симметрично расположенные абсциссы $x_1=x - ε/2 = (Ni+Ki - ε)/2$ и $x_2= x+ ε/2 = (Ni+Ki+ ε)/2$;
4. В полученных точках x_1 и x_2 оси X вычисляются (с помощью подпрограммы) два значения исходной функции $F_2(x_1)$ и $F_2(x_2)$;
5. Проверяется условие $F_2(x_1) > F_2(x_2)$. Если оно выполняется, то оператором присваивания $Ki=x_2$ производится уменьшение интервала поиска (за счёт сдвига влево его правой границы Ki), а его новое значение |Ki-Ni| присваивается переменной S. Если оно не выполняется, то оператором присваивания $Ni=x_1$ производится также уменьшение интервала поиска (теперь за счёт сдвига вправо его левой границы Ni), и его новое значение |Ki-Ni| присваивается переменной S;
6. Выводятся для просмотра пользователем текущие значения вычисленных переменных Ni, x_1, x_2, Ki, S, $F_2(x_1)$ и $F_2(x_2)$;
7. Проверяется условие окончания поиска |Ki-Ni|<2ε. Если оно не выполняется, то осуществляется переход к п.3 для очередного сужения интервала поиска. Если же оно выполняется, то вычисляются и выводятся для просмотра окончательные результаты поиска $x_m=(x_1+x_2)/2$ и $F_2(x_m)$. Схема алгоритма представлена на рис.1.

Составленная по такому алгоритму программа (см. приложение) выдаёт промежуточные, приведенные в таблице 1, и конечные результаты.

Таблица 1

	Поиск экстремума функции $F_2(x)$ методом дихотомии							
	Ni	x_1	x_2	Ki	Ki-Ni	$F_2(x_1)$	$F_2(x_2)$	*
	3. 7495	3. 7495	3. 7505	4. 0000	0. 2505	0. 878 185	0. 878 190	f 1<f2
	3. 7495	3. 8742	3. 8753	3. 8753	0. 1258	0. 878 485	0. 878 484	f 1>f2
	3. 8119	3. 8119	3. 8129	3. 8753	0. 0634	0. 878 432	0. 878 434	f 1<f2
	3. 8431	3. 8431	3. 8441	3. 8753	0. 0322	0. 878 483	0. 878 484	f 1<f2
	3. 8587	3. 8587	3. 8597	3. 8753	0. 0166	0. 878 490	0. 878 490	f 1=f2
	3. 8587	3. 8665	3. 8675	3. 8675	0. 0088	0. 878 489	0. 878 488	f 1>f2
	3. 8587	3. 8626	3. 8636	3. 8636	0. 0049	0. 878 490	0. 878 489	f 1>f2
	3. 8587	3. 8606	3. 8616	3. 8616	0. 0029	0. 878 490	0. 878 489	f 1>f2
	3. 8596	3. 8596	3. 8606	3. 8616	0. 0020	0. 878 490	0. 878 490	f 1=f2

x_m=3, 860 131 025, $F_2(x_m)$=0, 878 489 971.

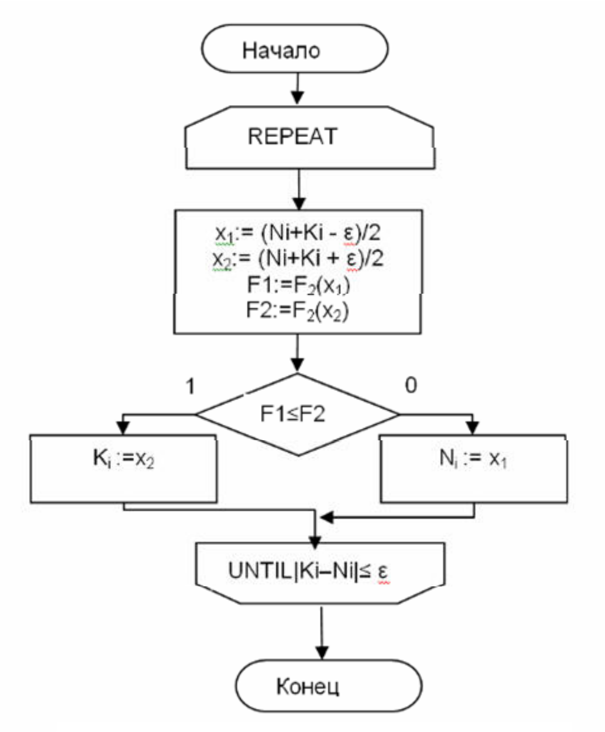


Рис.1. Схема алгоритма оптимизации по методу дихотомии

Контрольные вопросы ПЗ5(ОПК-4):

1. В чём заключается задача численного интегрирования?
2. Как группируются численные методы интегрирования?
3. Структурные схемы алгоритмов заданных методов прямоугольников?
4. Структурная схема алгоритма метода трапеций?
5. Структурная схема алгоритма метода Симпсона?
6. Вычисление интеграла при помощи пакета Excel?
7. Вычисление интеграла при помощи пакета MathCad?
8. Правило Рунге оценивания погрешности вычисления интеграла?
9. Сравнение методов численного интегрирования?

Практическое занятие №6. Структурные схемы алгоритмов, программирование и решение задач на ПК, связанных с проверкой условия унимодальности функции, выбором отрезка оптимизации и поиска точки её экстремума методами дихотомии и золотого сечения.

Поиск экстремума методом золотого сечения

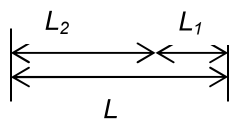


Рис.2. Деление отрезка длиной L

Метод золотого сечения является другим возможным способом стягивания отрезка поиска $[N_i; K_i]$ к точке экстремума. В отличие от метода дихотомии (метода деления отрезка пополам) в данном методе отрезок длиной $L = K_i – N_i$ разбивается точкой x_i на две неравные L_1 и L_2 части, где $L_1 < L_2$ и $L_1 + L_2 = L$.

Из бесчисленного множества возможных сечений отрезка по этому правилу только одно из них является и называется золотым. Для него должно быть справедливым равенство отношений:

$$\frac{L_1}{L_2} = \frac{L_2}{L} = \tau$$

Если рассматривать отрезок как единичный, т.е. L=1, отношение принимает вид:

$$\frac{1-L_2}{L_2} = \frac{L_2}{1}.$$

После преобразования отношение принимает вид:

$$L_2^2 + L_2 - 1 = 0.$$

Положительный корень их этого уравнения:

$$L_2 = \frac{-1+\sqrt{5}}{2} \approx 0{,}618,$$

а, соответственно,

$$\mathrm{L}_1 = 1 - L_2 = \frac{3-\sqrt{5}}{2} \approx 0{,}382, \qquad \tau = \frac{L_2}{L_1} \approx 0{,}618 \cdot$$

Но так как значение длины реального отрезка L практически всегда отличается от единицы, то истинные размеры его большей и меньшей частей могут определяться соответственно из выражений $L_2=\tau*(Ki–Ni)$ и $L_1=(1–\tau)*(Ki–Ni)$. При этом координата x_i точки сечения отрезка поиска на оси абсцисс зависит от порядка расположения на нём его частей. Ею будет либо $X_1=Ni+L_1=Ni+(1–\tau)(Ki–Ni)$, если первой располагается меньшая часть L_1, либо $X_2=Ni+L_2=Ni+\tau(Ki–Ni)$, если первой располагается большая часть L_2.

С учётом изложенного метод золотого сечения может быть реализован алгоритмом поиска экстремума (например, минимума) вида:

1. Задаются границы Ni, Ki отрезка поиска, погрешность вычисления ε и коэффициенты a_0, a_1 и a_2 функции $F_2(x)$;
2. Составляется подпрограмма вычисления $F_2(x)$ для задаваемых аргументов x;
3. Определяются абсциссы $x_2=Ni + \tau *(Ki – Ni)$ и $x_1=(1- \tau)*(Ki – Ni)$ точек сечения отрезка поиска минимума;
4. Вычисляются с помощью подпрограммы значения $F_2(x_1)$ и $F_2(x_2)$;
5. Проверяется условие $F_2(x_1) > F_2(x_2)$.

Если оно выполняется (что свидетельствует об убывании $F_2(x)$ в пределах от x_1 до x_2 и следовательно о нахождении минимума справа от $F_2(x_1)$), то операторами присваивания $Ni:=x_1$, $x_1:=x_2$ и $F_2(x_1) :=F_2(x_2)$ производится уменьшение интервала поиска (за счёт сдвига вправо в точку x_1 его левой границы Ni) и присвоение прежним значениям x_2 и $F_2(x_2)$ новых имён x_1 и $F_2(x_1)$. Затем для нового уменьшенного интервала поиска определяется новое значение x_2 и вычисляется новое значение $F_2(x_2)$.

Если это условие не выполняется (что свидетельствует о возрастании $F_2(x)$ в пределах от x_1 до x_2 и следовательно о нахождении минимума слева от x_2), то операторами присваивания $Ki:=x_2$, $x_2:=x_1$ и $F_2(x_2) :=F_2(x_1)$ производится уменьшение интервала поиска (за счёт сдвига влево в точку x_2 его правой границы Ki) и присвоение прежним значениям x_1 и $F_2(x_1)$ новых имён x_2 и $F_2(x_2)$. Затем для нового уменьшенного интервала поиска определяется x_1 и вычисляется $F_2(x_1)$.

6. Выводятся для просмотра пользователем текущие значения вычисленных на данном шаге итерации переменных x_1, Ni, x_2, Ki, $\varepsilon=|Ki - Ni|$, $F_2(x_1)$ и $F_2(x_2)$.

7. Проверяется условие окончания поиска $|Ki - Ni| < \varepsilon$. Если оно не выполняется, то осуществляется переход к п.5 для очередного стягивания отрезка поиска к точке минимума. В противном случае вычисляются и выводятся для просмотра окончательные результаты поиска $x_m= (Ki - Ni)/2$ и $F_2(x_m)$.

Составленная по приведенному алгоритму (рис.3) программа (см. приложение) выдаёт для исходной функции промежуточные и итоговые результаты, приведенные в таблице 2

Таблица 2

n	Поиск экстремума функции $F_2(x)$ методом золотого сечения							
	x_1	Ni	x_2	Ki	Ki-Ni	$F_2(x_1)$	$F_2(x_2)$	*
0	3. 8090	3. 6910	3. 8820	4. 0000	0. 3090	0. 878 425	0. 878 478	f1<f2
1	3. 8820	3. 8090	3. 9271	4. 0000	0. 1910	0. 878 478	0. 878 378	f1>f2
2	3. 8541	3. 8090	3. 8820	3. 9271	0. 1180	0. 878 489	0. 878 478	f1>f2
3	3. 8369	3. 8090	3. 8541	3. 8820	0. 0729	0. 878 477	0. 878 489	f1<f2
4	3. 8541	3. 8369	3. 8647	3. 8820	0. 0451	0. 878 489	0. 878 490	f1<f2
5	3. 8647	3. 8541	3. 8713	3. 8820	0. 0279	0. 878 489	0. 878 487	f1>f2
6	3. 8607	3. 8541	3. 8647	3. 8713	0. 0172	0. 878 490	0. 878 489	f1>f2
7	3. 8582	3. 8541	3. 8607	3. 8647	0. 0106	0. 878 489	0. 878 490	f1<f2
8	3. 8607	3. 8582	3. 8622	3. 8647	0. 0066	0. 878 490	0. 878 489	f1>f2
9	3. 8597	3. 8582	3. 8607	3. 8622	0. 0041	0. 878 490	0. 878 490	f1=f2

x_m=3. 860 199 928 и $F_2(x_m)$=0. 878 489 971.

Таким образом, точка максимума заданной функции $F_2(x)$ на плоскости X,Y определяется её координатами

$$x_m= 3.\ 860\ 131\ 025,\quad F_2(x_m) = 0.\ 878\ 489\ 971,$$

вычисленными по методу дихотомии, и координатами

$$x_m= 3.\ 860\ 199\ 928,\quad F_2(x_m) = 0.\ 878\ 489\ 971,$$

вычисленными по методу золотого сечения. Полученные результаты являются примерно одинаковыми. Однако второй метод считается более экономичным, так как на каждой итерации требуется вычисление только одного значения исходной функции, а не двух, как в методе дихотомии.

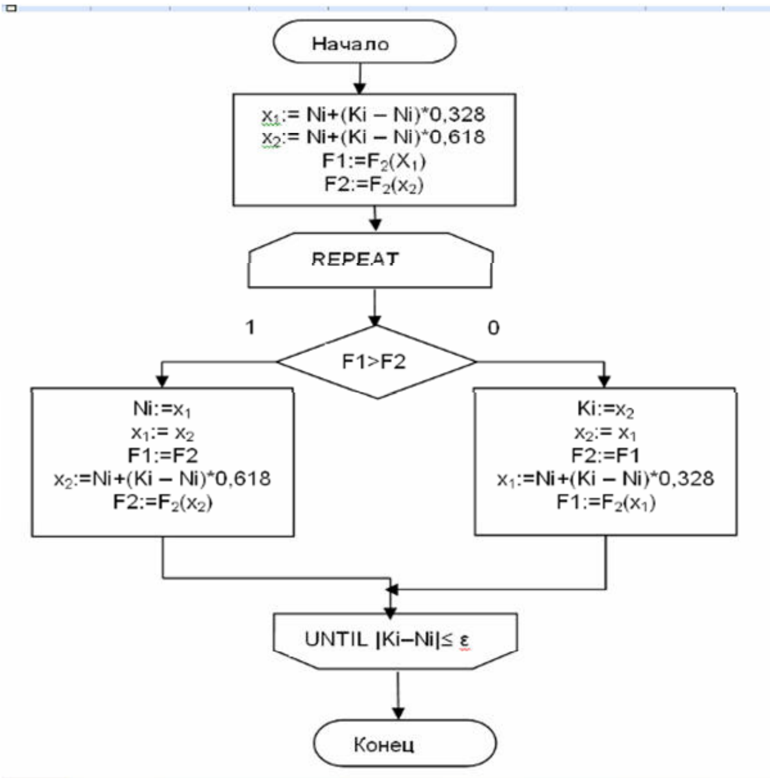


Рис..3. Схема алгоритма оптимизации по методу золотого сечения

Контрольные вопросы ПЗ6(ОПК-4):

1. Представление алгоритма и программы решения задачи одномерной оптимизации в соответствии с четвёртым заданием курсовой работы.
2. Выполнение программы на ПК. Отработка программы
3. Решение с помощью математических пакетов.
4. Сравнение результатов. Подготовка отчёта.

Лабораторная работа №4. Методы решения нелинейных уравнений. Составление алгоритма и программы решения. Решение задания средствами математических пакетов.

3. Вычисление интеграла методом прямоугольников

В интегрировании по методу прямоугольников предусматривается замена функции $F_2(x)$ полиномом нулевой степени $\varphi(x)=\beta_0 x^0=\beta_0$. При этом постоянная величина β_0 может приниматься равной значению подынтегральной функции в начале $\beta_0=F_2(x_i)$, в конце $\beta_0=F_2(x_{i+1})$ или в середине $\beta_0=F_2(x_{cp})$ каждого отрезка разбиения длиной h, где аргумент $x_{cp}=(x_i + x_{i+1}) /2$. При этом площадь каждого прямоугольника равна произведению его сторон $S_i = \beta_0 * h$, а сумма площадей всех n прямоугольников равна значению интеграла.

С учётом изложенного для вычисления интеграла по методу средних прямоугольников может быть использована формула:

$$I = \frac{X_2^* - X_1^*}{n} \sum_{i=0}^{n-1} F_2(Xi\,cp) = h \sum_{i=0}^{n-1} F_2(Xi + 0.5h)$$

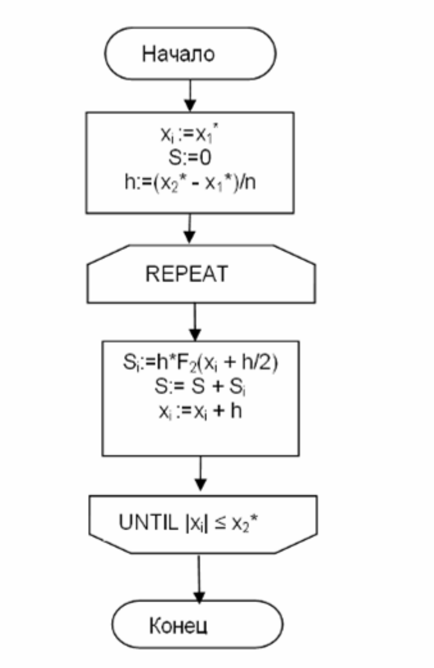


Рис.1. Схема алгоритма процесса интегрирования по методу средних прямоугольников

Точность интегрирования по этой формуле зависит от числа n отрезков разбиения. Она увеличивается с ростом n до его оптимального числа $n_{опт}$, при котором очевидный рост объёма вычислений приводит к росту погрешности за счёт округлений. Результаты программного (см. приложение) вычисления исходного интеграла по приведенной выше формуле для $n_1=10$ и $n_2=20$ приведены в таблице 10.1. Схема алгоритма вычислений представлена на рис.10.1.

4. Вычисление интеграла методом трапеций

В интегрировании по методу трапеций предусматривается замена функции $F_2(x)$ или её отдельных участков интегральным полиномом первой степени $\varphi(x)=\beta_0 + \beta_1 x$. Так как графиком такого полинома является прямая с угловым коэффициентом β_1, то предусматриваемая замена приводит к образованию трапеций с одинаковыми высотами $\mathbf{h=(x_2^*-x_1^*)/n}$ и чаще всего с разными основаниями $\mathbf{F_2(x_i)}$ и $\mathbf{F_2(x_{i+1})}$. Поскольку площадь одной трапеции равна произведению полусуммы оснований на высоту, и каждое основание, кроме внешних $\mathbf{F_2(x_0)=F_2(x_1^*)}$ и $\mathbf{F_2(x_n)=F_2(x_2^*)}$, принадлежит двум соседним трапециям, то для вычисления интеграла по этому методу может быть использована формула:

$$I = 0.5\,h\,[F_2(x_0) + F_2(x_n) + 2\sum_{i=1}^{n-1} F_2(x_i)]\,.$$

Точность вычислений по этой формуле выше точности вычисления интеграла по методу прямоугольников и также увеличивается с ростом n до $n_{опт}$, Результаты программного вычисления интеграла по этой формуле для $n_1=10$ и $n_2=20$ см. в таблице 1. Схема алгоритма предполагает циклическое накопление значений функции от 1-ой до (n-1)-ой с последующим вычислением значений функции на граничных значениях интервала интегрирования.

5. Вычисление интеграла методом Симпсона

В интегрировании по методу Симпсона предусматривается замена подынтегральной функции $F_2(x)$ или её парных участков интегральным полиномом второй степени $\varphi(x)=\beta_0 + \beta_1 x + \beta_2 x^2$. Так как графиком такого полинома является квадратичная парабола, то предусматриваемая замена приводит к образованию ***n/2*** криволинейных (с высотой ***2h***) трапеций. Поскольку парабола $\varphi(x)$ проходит через три соседние ординаты, т.е. точки плоскости X,Y с координатами первой из них x_i, $F_2(x_i)$, второй x_i+h, $F_2(x_i+h)$ и третьей x_i+2h, $F_2(x_i+2h)$, то можно доказать, что площадь одной такой трапеции $S_i=h\,[F_2(x_i) + 4\,F_2(x_i+h) + F_2(x_i+2h)]/3$, а сумму площадей всех криволинейных трапеций можно рассчитать по формуле

$$I=h[F_2(x_0)+F_2(x_n)+4(F_2(x_1)+F_2(x_3)+\ldots+F_2(x_{n-1}))+2(F_2(x_2)+F_2(x_4)+\ldots+F_2(x_{n-2}))]/3$$

Точность интегрирования по этой формуле существенно выше, чем по ранее приведенным формулам интегрирования по методам трапеций и прямоугольников. Результаты вычисления исходного интеграла по ней для $\boldsymbol{n_1=10}$ и $\boldsymbol{n_2=20}$ приведены в таблице 1. Схема алгоритма строится по аналогии с алгоритмом интегрирования по методу трапеций.

Таблица 1

n	Результаты интегрирования по методам		
	прямоугольников	трапеций	Симпсона
10	6, 873 993 874	6, 873 993 397	6, 943 428 993
20	6, 926 070 690	6, 926 070 213	6, 943 428 516

Суммарную погрешность вычисления интеграла в [1] предлагается оценить по правилу Рунге как сумму погрешностей выбранного метода и округлений:

$$R=\frac{\left|I_h-I_{h/2}\right|}{2^k-1},$$

где - I_h и $I_{0,5h}$ - значения интегралов при n=10 и n=20 (из первой и второй строк таблицы .1),

- k - принимается равным 2-м для метода средних прямоугольников и метода трапеций и равным 4-м для метода Симпсона. С учётом изложенного получаем:

$R_{ср.\ пр} = 0,\ 017\ 358\ 938,$

$R_{тр} = 0,\ 017\ 358\ 938,$

$R_{Сип} = 0,\ 000\ 000\ 032.$

Соответственно, наименьшую погрешность имеет метод Симпсона, т.к. интерполирующая функция второго порядка, построенная по трем точкам функции $F_2(x)$ всегда точнее приближается к исходной функции.

Контрольные вопросы ЛР4(ОПК-4):

1. Что является корнем нелинейного уравнения?
2. Из каких этапов состоит решение нелинейного уравнения?
3. Как производится этап отделения корней?
4. На какой теореме основывается аналитическое отделение корней?
5. Что является условием окончания поиска корня?
6. В чём суть метода простой итерации уточнения корня?
7. Какие условия должны быть выполнены, чтобы процесс сходился к корню?
8. Сравните методы уточнения корней.
9. Приведите структурную схему алгоритма заданного метода, уточняющего корень нелинейного уравнения.
10. Представление алгоритма и программы решения нелинейного уравнения в соответствии со вторым заданием курсовой работы.
11. Выполнение программы на ПК. Отработка программы. Решение с помощью математических пакетов.
12. Сравнение результатов. Подготовка отчёта.

Лабораторная работа №5. Выполнение программы на ПК на основе методов численного интегрирования решения задач. Сравнение с результатами решения с помощью математических пакетов.

1.Вычисление интеграла при помощи пакета Excel

В Excel нет стандартных функций для вычисления интегралов. Поэтому расчеты должны организовываться как последовательное вычисление нескольких шагов. Количество шагов n определяется произвольно или исходя из требуемой точности из следующих соотношений:

- для метода прямоугольников:

$$n > \max_{[x_1^*, x_2^*]} \left| F_2'(x) \right| \cdot \frac{(x_2^* - x_1^*)^2}{\varepsilon},$$

- для метода трапеций:

$$n > \sqrt{\max_{[x_1^*, x_2^*]} \left| F_2''(x) \right| \cdot \frac{(x_2^* - x_1^*)^3}{6\varepsilon}},$$

- для метода Симпсона:

$$n > \sqrt[4]{\max_{[x_1^*, x_2^*]} \left| F_2''''(x) \right| \cdot \frac{(x_2^* - x_1^*)^5}{90\varepsilon}}.$$

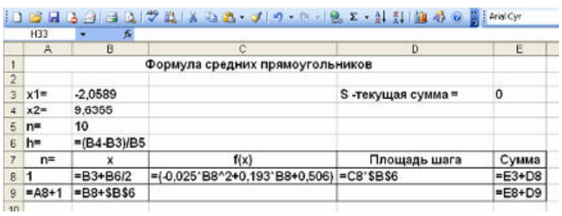
где - $F_2(x)$ – функция для интегрирования, для которой определяются соответствующие производные. В рассматриваемой задаче это функция квадратичной аппроксимации;

- x_1^*, x_2^* - границы интегрирования, являющиеся в этой задаче корнями функции $F_2(x)$;
- ε – требуемая точность;

Для рассматриваемого задания расчетные формулы показаны на рис.2.

В диапазоне A3:B6 показаны константы, использующиеся при расчетах. В ячейке E3 определено начальное значение суммы элементарных прямоугольников, из которых будет складываться площадь фигуры. Эта цифра будет пополняться на каждом шаге (E8) прибавлением площади элементарной фигуры, вычисленной на этом шаге – D8. Ячейка B8 показывает первое значение переменной ***x***, при которой рассчитывается высота прямоугольника. Последующие координаты x нарастают на каждом шаге прибавлением величины шага h.

Рис.2 показывает начальные не всегда повторяющиеся формулы для расчета. Свободные ячейки строкиA9:E9 должны быть заполнены копированием выше расположенных строк одноименных столбцов. После заполнения строка A9:E9 будет готова для копирования методом автозаполнения для получения 10 шагов расчетов. В ячейке E17 будет выведено значение интеграла для 10 шагов - 6,973324621.



	A	B	C	D	E
1			Формула средних прямоугольников		
2					
3	x1=	-2,0589		S -текущая сумма =	0
4	x2=	9,6355			
5	n=	10			
6	h=	=(B4-B3)/B5			
7	n=	x	f(x)	Площадь шага	Сумма
8	1	=B3+B6/2	=(-0,025*B8^2+0,193*B8+0,506)	=C8*$B$6	=E3+D8
9	=A8+1	=B8+$B$6			=E8+D9

Рис..2. Вычисление интеграла методом прямоугольников

2. Вычисление интеграла при помощи пакета MathCad

Для вычисления интеграла используется оператор определенного интегрирования, который имеет поля для ввода значений начала и конца интервала интегрирования. Кроме того, значение интеграла может быть получено в символьном виде.

Для получения доступа к оператору определенного интегрирования необходимо на панели инструментов нажать кнопку ***Операторы математического анализа*** и выбрать оператор из появившейся панели (см. рис.3).

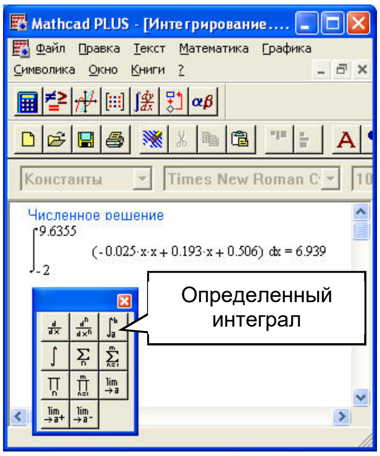


Рис.3. Вычисление определенного интеграла

Организовать вычисления возможно иначе: сначала определить функцию, а затем вписать ее обозначение в подинтегральное выражение.

Контрольные вопросы ЛР5(ОПК-4):

1. Представление алгоритма и программы численного интегрирования в соответствии с третьим заданием курсовой работы.
2. Выполнение программы на ПК. Отработка программы
3. Решение с помощью математических пакетов.
4. Сравнение результатов. Подготовка отчёта.

Лабораторная работа №6. Методы одномерной оптимизации решения задач. Составление алгоритма и программы решения. Решение задания средствами математических пакетов.

5. Поиск экстремума с помощью пакета Excel

Поиск экстремума с помощью пакета Excel предполагает построение расчетных формул в соответствии с используемым алгоритмом. Для метода дихотомии расчетные формулы представлены на рис.4.

Строка A6:K6 использует константы, указанные выше таблицы. Последующая строка A7:K7 после копирования в нее недостающих формул из предшествующей строки одноименного столбца будет готова для копирования целиком методом автозаполнения.

	A	B	C	D	E	F
1				Opt по методу дихотомии		
2	ε =	0,001				
3	Ni =	3				
4	Ki =	5				
5	N1	K1	x1	x2	F2(x1)	F2(x2)
6	=B3	=B4	=0,5*(B3+B4-B2/2)	=0,5*(B3+B4+B2/2)	=-(-0,0255*C6^2+0,1932*C6+0,5059)	=-(-0,0255*D6^2+0,1932*D6+0,5059)
7	=G6	=H6	=0,5*(A7+B7-$B$2/2)	=0,5*(B7+C7+$B$2/2)		
8						

	G	H	I	J	K
1					
2					
3					
4					
5	N2	K2	|K2-N2|	Min=	Прим
6	=ЕСЛИ(E6<F6;A6;C6)	=ЕСЛИ(E6<F6;D6;B6)	=ABS(H6-G6)	=ЕСЛИ(I6<$B$2;(D6+C6)/2	=ЕСЛИ(I6<$B$2;"stop";"go")
7					
8					

Рис..4. Формулы для поиска минимума по методу дихотомии

Следует учесть, что точность вычисления и округления в машине потребует некоторого изменения расчетных формул. В частности, для переменных X1 и X2 следует отступить от средней точки влево и вправо не величину точности, а ее половину (см.рис.4). В противном случае сближение граничных точек Ni и Ki никогда не будет меньше или равно величине точности. Это получится потому, что сближение меньше точности не может быть в соответствии с видом самой формулы расчета X1 и X2, а равенство не будет получено из-за механизмов округления в машине.

Кроме того, важно помнить, что условные формулы расчета новых значений отрезка N2 и K2 в совокупности являются эквивалентом условного оператора (см. схему алгоритма на рис.1). Однако каждая из них вычисляет только одну переменную: либо N2, либо K2. Поэтому каждая формула либо по условию изменяет ее, либо оставляет прежнее значение только своей переменной.

Результаты расчета по приведенным формулам представлены на рис.5.

По значениям в ячейка A6:B23 видно как сжимается отрезок приближения, и с какой стороны это происходит. По величине самого отрезка (I6:I23) наглядно видно как уменьшение происходит до величины меньше точности (0,001). Как только точность превысит заданную (ε≤0,001) в ячейке I23, индикатор фиксирует ST□□ и появляется искомое значение x_{min}=3,7889.

	A	B	C	D	E	F	G	H	I	J	K
1		Opt по методу дихотомии									
2	ε =	0,001									
3	Ni =	3									
4	Ki =	5									
5	N1	K1	x1	x2	F2(x1)	F2(x2)	N2	K2	|K2-N2|	Min=	Прим
6	3	5	3,99975	4,00025	-0,8707	-0,8707	3	4,00025	1,00025		go
7	3	4,00025	3,499875	3,750313	-0,86972	-0,87181	3,499875	4,00025	0,500375		go
8	3,499875	4,00025	3,7498125	3,875281	-0,87181	-0,87165	3,499875	3,87528125	0,37540625		go
9	3,499875	3,8752813	3,687328125	3,781555	-0,87158	-0,87184	3,68732813	3,87528125	0,18795313		go
10	3,687328	3,8752813	3,781054688	3,828418	-0,87184	-0,8718	3,68732813	3,828417969	0,14108984		go
11	3,687328	3,828418	3,757623047	3,793271	-0,87182	-0,87184	3,75762305	3,828417969	0,07079492		go
12	3,757623	3,828418	3,792770508	3,810844	-0,87184	-0,87183	3,75762305	3,810844238	0,05322119		go
13	3,757623	3,8108442	3,783983643	3,797664	-0,87184	-0,87184	3,75762305	3,79766394	0,04004089		go
14	3,757623	3,7976639	3,777393494	3,787779	-0,87184	-0,87184	3,77739349	3,79766394	0,02027045		go
15	3,777393	3,7976639	3,787278717	3,792721	-0,87184	-0,87184	3,77739349	3,792721329	0,01532784		go
16	3,777393	3,7927213	3,784807411	3,789014	-0,87184	-0,87184	3,78480741	3,792721329	0,00791392		go
17	3,784807	3,7927213	3,78851437	3,790868	-0,87184	-0,87184	3,78480741	3,790867849	0,00606044		go
18	3,784807	3,7908678	3,78758763	3,789478	-0,87184	-0,87184	3,78480741	3,78947774	0,00467033		go
19	3,784807	3,7894777	3,786892576	3,788435	-0,87184	-0,87184	3,78689258	3,78947774	0,00258516		go
20	3,786893	3,7894777	3,787935158	3,788956	-0,87184	-0,87184	3,78689258	3,788956449	0,00206387		go
21	3,786893	3,7889564	3,787674512	3,788565	-0,87184	-0,87184	3,78767451	3,788956449	0,00128194		go
22	3,787675	3,7889564	3,78806548	3,788761	-0,87184	-0,87184	3,78767451	3,788760965	0,00108645		go
23	3,787675	3,788761	3,787967738	3,788614	-0,87184	-0,87184	3,78767451	3,788614351	0,00093984	3,788291	stop
24											

Рис.5. Результаты расчета экстремума по методу дихотомии

6. Поиск экстремума с помощью пакета MathCad

При использовании пакета MathCad для минимума унимодальной функции от одной переменной на заданном отрезке используется функция ***Minerr(x)***. Функция возвращает значение x, минимизирующее расхождение с начальным приближением. Учитываются все уравнения и неравенства, расположенные между ключевым словом Given и функцией Minerr.

Для вычисления требуется задание начального приближения для поиска (x=3,4 для рассматриваемого примера) и описание исходной функции и целевой функции. Поскольку минимум функции ищется по равенству нулю ее первой производной, то она и является ее целевой функцией (условием минимизации). Это условие задается после начала вычислительного блока (***Given***). Все остальные неравенства (диапазон поиска x>3 и x<5) также должны располагаться между словом Given и функцией Minerr.

Рабочее поле пакета для поиска минимума показано на рис.6.

Если пакет не может вычислить экстремум, часто это указывает на то, что следует уточнить начальное приближение.

Для найденного экстремума построен график на основе заданного диапазона переменных (см.рис.6). Однако следует помнить, что исходная функция $F_2(x)$ на самом деле имеет максимум. Поэтому график является зеркальным отображение по отношению к реальной кривой. Он получен за счет добавление знака отрицания в выражение $F_2(x)$ для обеспечения условий применения функции Minerr. Реальное значение максимального значения функции $F_2(x)$= 0,872.

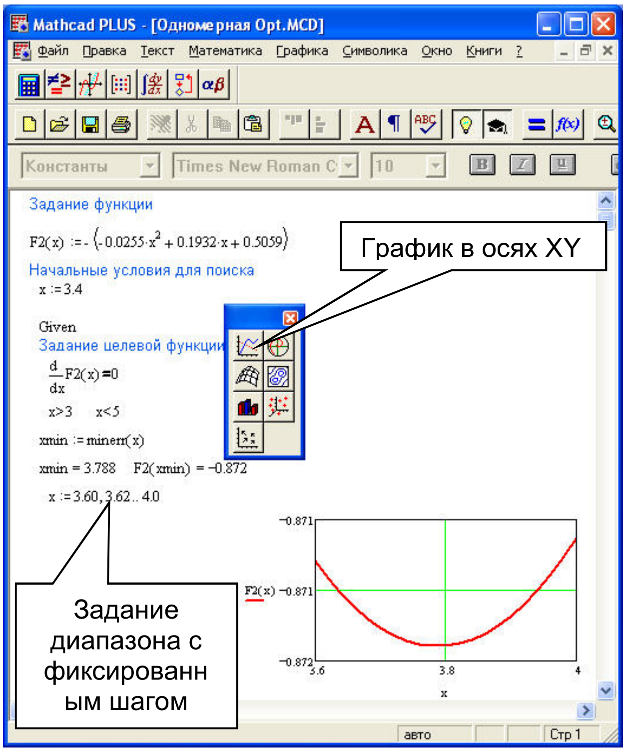


Рис.6. Выражения для расчета минимума

Вторая возможность – поиск нулей первой производной, то есть стандартный математический подход. Для этого можно использовать блок ***Given – Find***. Функция Find ищет точное решение системы уравнений и неравенств, записанных после слова given.

В старших версиях Mathcad'a появилась дополнительная возможность поиска экстремумов с помощью функций ***Minimize*** и ***Maximize***, которые могут быть использованы как сами по себе, так и совместно с блоком given. Аргументы функций: имя функции, экстремум которой ищется, и список ее аргументов.

Контрольные вопросы ЛР6(ОПК-4):

1. Задача оптимизации?
2. Цель одномерного поиска ?

3. Когда задача имеет единственное решение?
4. Какая функция называется унимодальной?
5. Критерии унимодальности функции?
6. В чём заключается метод прямого перебора нахождения экстремума функции?
7. Структурная схема алгоритма метода перебора?
8. В чём заключается метод последовательного перебора (метод дихотомии) нахождения экстремума функции?
9. Структурная схема алгоритма метода дихотомии?
10. В чём заключается метод последовательного перебора (метод золотого сечения) нахождения экстремума функции?
11. Структурная схема алгоритма метода золотого сечения?
12. Сравнение заданных методов нахождения экстремума функции?
13. Определение экстремума функции средствами пакета Excel и пакета MathCad.

ПРИЛОЖЕНИЕ 1.

Примеры возможных программ решения.

1. Аппроксимация.

1. Программа определения параметров линейной аппроксимации.

```
rogram linar;
USES CRT;
Const
X: array[0..5] of real=(-11,-9,-7,-5,-3,-1);
Y: array[0..5] of real=(1.256, 0.911, -0.436, -1.416, -1.03, -0.356);
n:integer=6;
 Var
Ro,Sx,Sx2,Sy,Sxy,a0,a1: real;
i: integer;
F1:array[0..5] of real;
          BEGIN
         CLRSCR;
Sx:=0;   Sx2:=0;
Sy:=0;    Sxy:=0;
Ro:=0;
        For i: = 0 to 5 do
Begin
        Sx:=Sx + X[i];
        Sy:=Sy + Y[x];
        Sxy:=Sxy + X[i] * Y[i];
        Sx2:=Sx2 + sqr(X[i]);
End;
       Sx:=Sx/n;
       Sy:=Sy/n;
       a1:= (Sxy – n*Sx*Sy)/(Sx2 – n*sqr(Sx));
       a0:= Sy – a1*Sx;
Writeln(‘Коэффициенты линейной аппроксимации’);
Writeln(‘ a0 =’, a0:6:4, ‘a1=’, a1:6:4);
Writeln(‘ Значения линейной функции в узлах аппроксимации’);
        For i:=0 to 5 do
Begin
       F1[i]:= a0 + a1*X[i];
       Ro:= Ro + sqr((F1[i] – Y[i]));
Writeln(‘ F1(X’ , i , ‘ )=’, F1[i]:8:4;
End;
Ro:=sqrt(Ro/n);
Writeln( ‘Качество аппроксимации ‘, Ro:4:4);
Readln;
END.
```

2. Программа построения матрицы Грамма и решения системы нормальных уравнений.

.

```
rogram kvadappr;
uses crt;
const
X:array[0..5] of real=(5, 7, 9, 11, 13, 15);
Y:array[0..5] of real=(-0.116, -0.876, -0.19, 1.576, 3.056, 3.25);
```

```
n:integer=6;
var
Ro,Sx0,Sx,Sx2,Sx3,Sx4,Sy,Sxy,Sx2y,a0,a1,a2:real;
C,M:real;
i,j,k,L:integer;
F2:array[0..5] of real;
Sys:array[1..3,1..4] of real;
   begin
   CLRSCR;
   Sx0:=0; Sx:=0; Sx2:=0; Sx3:=0; Sx4:=0;
   Sy:=0; Sxy:=0; Sx2y:=0;
   writeln('|i| x^0 | x | x^2 | x^3 | x^4 |  y   | y*x  |   y*x^2  | ');
   writeln('____________________________________________________________');
   for i:=0 to 5 do
 begin
 Sx0:=Sx0+1;
 Sx:=Sx+X[i];
 Sx2:=Sx2+sqr(X[i]);
 Sx3:=Sx3+X[i]*X[i]*X[i];
 Sx4:=Sx4+sqr(sqr(X[i]));
 Sy:=Sy+Y[i];
 Sxy:=Sxy+X[i]*Y[i];
 Sx2y:=Sx2y+X[i]*X[i]*Y[i];
 write('|',i,'| ',1,'   |',X[i]:3:0,'|',X[i]*X[i]:4:0,' |',X[i]*X[i]*X[i]:4:0,'    |',sqr(X[i]*X[i]):5:0);
 writeln('|',Y[i]:6:3,'|',X[i]*Y[i]:7:3,'|',sqr(X[i])*Y[i]:8:3,' |');
 end;
    writeln('_____________________________________________________');
   write('| S |',Sx0:1:0,' |',Sx:3:0,'|',Sx2:4:0,'|',Sx3:4:0,'|',Sx4:5:0);
    writeln('|',Sy:6:3,'|',Sxy:7:3,'|',Sx2y:8:3,' |');
    Sys[1,1]:=Sx0; Sys[1,2]:=Sx; Sys[1,3]:=Sx2; Sys[1,4]:=Sy;
    Sys[2,1]:=Sx;  Sys[2,2]:=Sx2;Sys[2,3]:=Sx3; Sys[2,4]:=Sxy;
    Sys[3,1]:=Sx2; Sys[3,2]:=Sx3;Sys[3,3]:=Sx4; Sys[3,4]:=Sx2y;
 {Решение системы нормальных уравнений}
      for i:=1 to 2 do
     begin
     C:=Sys[i,i];
    for j:=1 to 4 do
      begin
      Sys[i,j]:=Sys[i,j]/C;
     end;
    for k:=i+1 to 3 do
     begin
     M:=Sys[k,i];
    for L:=i to 4 do
      begin
      Sys[k,L]:=Sys[k,L]-Sys[i,L]*M;
```

```
        end;
        end;
        end;
        a2:=Sys[3,4]/Sys[3,3];
        a1:=Sys[2,4]-Sys[2,3]*a2;
        a0:=Sys[1,4]-Sys[1,3]*a2-Sys[1,2]*a1;
  writeln('Коэффиценты квадратичной аппроксимации:');
  writeln('a0=',a0:6:4,'','a1=',a1:6:4,'','a2=',a2:6:4);
  writeln('Значения квадратичной функции в узлах аппроксимации');
       for i:=0 to 5 do
        begin
        F2[i]:=a2*sqr(X[i])+a1*X[i]+a0;
        Ro:=Ro+sqr(F2[i]-Y[i]);
        writeln('F2(x',i,')=',F2[i]:10:4);
        end;
      Ro:=sqrt(Ro/n);
      writeln('Качество аппроксимации Ro=',Ro:4:4);
      readln;
      end.
```

2. Нелинейные уравнения.

```
□rogram ITERACIA;
uses crt;
const e=0.0001;
var
a,b,c,x,x0,fi,l,t:real;
n:integer;
function f(x:real):real;
begin
f:=a*sqr(x)+b*x+c;
end;
begin
clrscr;
n:=0;
writeln('vvedite A B C');
readln (a,b,c);
writeln('vvedite L');
readln(l);
writeln('vvedite nachalnoe pribligenie x');
readln(x);
repeat
x0:=x;
fi:=x0+l*f(x0);
x:=fi;
```

```
t:=abs(x-x0);
n:=n+1;
until (t<e) or (n=50);
writeln('iscomiy coren',x:8:3);
Readln;
end.



□rogram Nuton;
Uses Crt;
Var
a,b,c,x0,Xn,Xt,R,E:real;
n:integer;
Function F2(x:real):real;
Begin
F2:=a*x*x+b*x+c;
End;
Function F2p(x:real):real;
Begin
F2p:=2*a*x+b
End;
Begin
Writeln('Vvesti E=');
Readln(E);
n:=0;
Clrscr;
Writeln('Vvesti a,b,c');
Readln(a,b,c);
Writeln('Vvesti nachalnoe pribligenie x0=');
Readln(x0);
writeln('| n |',' Xn ','':5,' F2(Xn) ','':5,' F2p(Xn) ');
repeat
Writeln('|','':2,n  ,'':5,x0:8:4,'':7, F2(x0):8:5,'':7,F2p(x0):8:5);
Xn:=x0-F2(x0)/F2p(x0);
xt:=x0;
x0:=Xn;
R:=abs(xt-xn);
n:=n+1;
Until(abs(xt-xn)<E) or (n=20);
Writeln('utochnenni koren=',x0:8:4);
Readln;
End.
```

3.Численное интегрирование

```
□R□GRAM SREDNIX□RIAM□YG;
USES CRT;
VAR
A,B,C,□v,□n,X,h,S,I:REAL;
n,k:INTEGER;
FUNCTI□N F2(x:REAL):REAL;
BEGIN
F2:=A*X*X+B*X+C;
END;
BEGIN
CLRSCR;
WRITE('BBEDITE A,B,C');
READLN(A,B,C);
WRITE('BBEDITE BERXNII I NIGNII □REDEL INTEGRIR □v,□n');
READLN(□v,□n);
WRITE('BBEDITE 4ISL□ □TREZK□B RAZBIENIA n');
READLN(n);
h:=(□v-□n)/n;
X:=□n;
S:=0;
F□R K:=1 T□ n D□
BEGIN
S:=S+F2(X+h/2);
X:=X+h;
END;
I:=h*S;
WRITELN('□RI n=',n,'□SK□MI INTEGRAL RAVEN',I:8:4);
READLN;
END.
```

```
□R□GRAM TRA□ECIA;
USES CRT;
VAR
A,B,C,□v,□n,X,h,S,I:REAL;
n,k:INTEGER;
FUNCTI□N F2(x:REAL):REAL;
BEGIN
F2:=A*X*X+B*X+C;
END;
BEGIN
CLRSCR;
WRITE('BBEDITE A,B,C');
READLN(A,B,C);
WRITE('BBEDITE BERXNII I NIGNII □REDEL INTEGRIR □v,□n');
```

```
READLN(□v,□n);
WRITE('BBEDITE 4ISL□ □TREZK□B RAZBIENIA n');
READLN(n);
h:=(□v-□n)/n;
X:=□n;
S:=0;
F□R K:=1 T□ n-1 D□
BEGIN
S:=S+F2(X);
X:=X+h;
END;
I:=0.5*h*(F2(□n)+F2(□v)+2*S);
WRITELN('□RI n=',n,'□SK□MI INTEGRAL RAVEN',I:8:4);
READLN;
END.


□R□GRAM SIM□S□N;
USES CRT;
VAR
A,B,C,X,X1,X2,h:REAL;
n,i,j:INTEGER;
S1,S2,Isim:REAL;
FUNCTI□N F2(X:REAL):REAL;
BEGIN
F2:=A*X*X+B*X+C;
END;
BEGIN
CLRSCR;
WRITE('BBEDITE A,B,C');
READLN(A,B,C);
WRITE('BBEDITE X1,X2');
READLN(X1,X2);
n:=0;
F□R i:=1 T□ 2 D□
BEGIN
n:=n+10;
h:=(X2-X1)/n;
S1:=0;
S2:=0;
X:=X1;
F□R j:=1 T□ n-1 D□
BEGIN
X:=X+h;
if odd(j) then
S1:=S1+F2(X)
```

```
else
S2:=S2+F2(X);
END;
Isim:=h*(F2(X1)+F2(X2)+4*S1+2*S2)/3;
WRITELN('n=',n,'Isim=',Isim:12:5);
END;
READLN;
END.
```

4. Одномерная оптимизация.

```
□rogram dihot;
UsesCrt;
Var
a,b,d,E:real;
x1,x2,F1,F2:real;
Xmin,Fmin:real;
Begin
Clrscr;
Writeln('Ввести a,b,d,E');
Readln(a,b,d,E);
  Repeat
begin
x1:=(a+b-d)/2;
x2:=(a+b+d)/2;
     F1:= 0.0155*x1*x1 + 0.6419*x1  + 3.2069;
     F2:= 0.0155*x2*x2 + 0.6419*x2 + 3.2069;
    If F1<F2 then b:=x2 else a:=x1;
end;
  Until abs(b-a)<=E;
Xmin:=(a+b)/2;
Fmin:=0.0155*Xmin*Xmin+0.6419*Xmin+ 3.2069;
Writeln('Xmin=',Xmin:10:3,'':5,'Fmin=',Fmin:10:3);
Readln;
     End.
```

```
□rogram Zol;
uses crt;
var
a,b,E:real;
x1,x2,xm,F1,F2,Fm,k1,k2:real;
        Begin
        Clrscr;
writeln('ввести a,b,E');
readln(a,b,E);
```

```
k1:=(3-sqrt(5))/2;
k2:=(sqrt(5)-1)/2;
x1:=a+k1*(b-a);
x2:=a+k2*(b-a);
F1:= 0.0596*x1*x1 - 1.9163*x1  + 12.2743;
F2:= 0.0596*x2*x2 - 1.9163*x2  + 12.2743;
while abs(b-a)>E do
     begin
if f1<f2 then
begin
b:=x2;
x2:=x1;
F2:=F1;
X1:=a+k1*(b-a);
F1:= 0.0596*x1*x1 - 1.9163*x1  + 12.2743;
 end
ELSE
begin
a:=x1;
x1:=x2;
F1:=F2;
X2:=a+k2*(b-a);
F2:= 0.0596*x2*x2 - 1.9163*x2  + 12.2743;
end
end;
xm:=(a+b)/2;
fm:=0.0596*Xm*Xm - 1.9163*Xm + 12.2743 ;
Writeln('Xmin=',Xm:10:3,'':5,'fmin=',fm:10:3);
readln;
End.
```