

Лабораторная работа №2
«Безусловная одномерная оптимизация»
(6 часов)

Цель работы: изучение различных методов одномерной оптимизации.

1 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

1.1 Постановка задачи одномерной безусловной оптимизации

Составной частью многих методов многомерной оптимизации является поиск экстремума функции одной переменной. Будем рассматривать задачу оптимизации на примере задачи минимизации, которая формулируется в виде:

$$\min f(x), \quad x \in X, \quad (1.1)$$

где X – множество допустимых точек, среди которых ищется точка x^* , доставляющая минимум целевой функции $f(x)$.

В общем случае, оптимизация предполагает поиск экстремального (минимального или максимального) значения функции. Однако задача (1.1) эквивалента задаче

$$\max(-f(x)), \quad x \in X, \quad (1.2)$$

что позволяет переносить решения, полученные для задачи минимизации на задачу максимизации, и наоборот.

Если X есть вещественная ось чисел и целевая функция $f(x)$ имеет только один аргумент, то имеет место задача одномерной безусловной минимизации.

В задачах проектирования не существует бесконечных величин. Поэтому решение ищется на некотором отрезке $X = [a, b]$, содержащем искомое решение x^* . Такой отрезок называется отрезком локализации. Относительно целевой функции $f(x)$ предполагается, что она унимодальная.

Определение 1.1: Функция $f(x)$ называется унимодальной на $X = [a, b]$, если существует такая точка $x^* \in X$, что

$$f(x_1) > f(x_2), \text{ если } x_1 < x_2 < x^*, \quad x_1, x_2 \in X,$$

$$f(x_1) < f(x_2), \text{ если } x^* < x_1 < x_2, \quad x_1, x_2 \in X.$$

В ряде методов относительно целевой функции $f(x)$ предполагается, что она выпуклая на X .

Определение 1.2: Функция $f(x)$ называется выпуклой на $X = [a, b]$, если $f(\alpha \cdot x_1 + (1 - \alpha)x_2) \leq \alpha \cdot f(x_1) + (1 - \alpha) \cdot f(x_2)$ при любых $x_1, x_2 \in X$ и любых α , $0 \leq \alpha \leq 1$.

Если при любых $x_1, x_2 \in X$ неравенство будет строгим, то функция $f(x)$ называется строго выпуклой. Непрерывная строго выпуклая функция является унимодальной. Однако не всякая унимодальная функция является выпуклой или непрерывной.

1.1.2 Алгоритмы пассивного и активного поиска минимума

Пусть $[a, b]$ – исходный отрезок локализации и пусть N – количество точек, в которых необходимо провести вычисления целевой функции $f(x)$. Точки, в которых необходимо провести вычисления, определяются следующим образом:

1) Если N – нечетное ($N = 2k - 1$), то

$$x_i = a + \frac{b-a}{N+1} \cdot i, \quad i = 1, 2, 3, \dots, N.$$

2) Если N – четное ($N = 2k$), то

$$x_{2i} = a + \frac{b-a}{k+1} \cdot i, \quad x_{2i-1} = x_{2i} - \delta, \quad \delta > 0, \quad i = 1, 2, 3, \dots, k.$$

Среди множества вычисленных значений $f(x_i)$ ищется точка x_j , в которой достигается минимум:

$$f(x_j) = \min\{f(x_i)\}, \quad i = \overline{1, N}.$$

Найденная точка принимается за приближенное решение задачи $\tilde{x} = x_j$.

Исходный отрезок локализации сужается до отрезка $[x_{j-1}, x_{j+1}]$, длина которого равна

$$L = x_{j+1} - x_{j-1},$$

причем

$$L_N = 2 \frac{b-a}{N+1}, \quad \text{если } N = 2k - 1,$$

$$L_N = 2 \frac{b-a}{k+1} + \delta, \quad \text{если } N = 2k.$$

Точность найденного решения \tilde{x} равна половине отрезка локализации, т.е.

$$|x^* - \tilde{x}| \leq \varepsilon,$$

где $\varepsilon = \frac{1}{2}L_N$, x^* – точное решение.

В алгоритмах активного поиска очередная точка, в которой происходит вычисление функции, выбирается с учетом информации, полученной на предыдущих этапах. Рассмотрение этих алгоритмов начнем с методов блочного поиска, которые сочетают в себе пассивные и последовательные стратегии поиска. При этом вычисления в точках объединены в блоки, в каждом из которых проводится одновременно n_i вычислений. Тогда общее число вычислений

$$N = \sum_{i=1}^m n_i.$$

Таким образом, блок – это совокупность из нескольких вычислений, которые проводятся одновременно (пассивный поиск).

Результаты, полученные в $(i-1)$ -м блоке, становятся известны перед началом вычислений в i -м блоке (x_{ij} , где $j = \overline{1, n_i}$). Это определяет последовательный поиск. Если размеры блоков равны единице, т.е. $n_i = 1$, то имеем обычный последовательный алгоритм поиска.

Алгоритм равномерного блочного поиска

Шаг_1. Задаются исходный отрезок локализации $[a, b]$, ε – точность приближенного решения \tilde{x} , число вычислений в блоке – n (нечетное, $n = 2k - 1$). Проводим вычисление функции в середине отрезка $[a, b]$, т.е. вычисляем

$$y_k = f(x_k), \text{ где } x_k = (a + b)/2.$$

Шаг_2. Вычисляем значение функции в остальных точках:

$$y_i = f(x_i),$$

$$\text{где } x_i = a + i \frac{(b-a)}{(n+1)}, \quad j = \overline{1, n_i}, \quad i \neq k.$$

Находим точку x_j , в которой достигается минимум среди вычисленных значений:

$$f(x_j) = \min\{f(x_i)\},$$

следовательно, точное значение минимума x^* содержится на отрезке $[x_{j-1}, x_{j+1}]$.

Шаг_3. Полагаем:

$$a = x_{j-1}, b = x_{j+1}, x_k = x_j, y_k = y_j.$$

Если $b - a \leq 2\varepsilon$, то $\tilde{x} = x_k$, $\tilde{y} = y_k$ и поиск заканчивается. Иначе перейти к шагу 2. Если заданная точность ε достигнута после m итераций, то длина отрезка локализации после всех N вычислений (где $N = n + (m-1)(n-1) = (n-1)m + 1$) будет:

$$L_N = \left(2 \frac{b-a}{n+1}\right)^m,$$

и

$$|x^* - \tilde{x}| \leq \frac{1}{2} L_N.$$

Алгоритм деления интервала пополам

Данный алгоритм является вариантом предыдущего алгоритма при $n = 3$.

Шаг 1. Задаются a, b, ε . Вычисляем значение функции в точке $x_2 = (a+b)/2$, т.е. находим $y_2 = f(x_2)$.

Шаг 2. Вычисляем значение функции в остальных точках блока:

$$x_1 = (a + x_2)/2, y_1 = f(x_1), x_3 = (x_2 + b)/2, y_3 = f(x_3).$$

Находим значение x_j из условия $f(x_j) = \min\{f(x_i)\}$, $i = 1, 2, 3$.

Тогда точное решение x^* содержится на отрезке $[x_{j-1}, x_{j+1}]$.

Предполагается $x_0 = a$, $x_4 = b$.

Шаг 3. Полагаем:

$$a = x_{j-1}, b = x_{j+1}, x_2 = x_j, y_2 = y_j.$$

Если $b - a \leq 2\varepsilon$, то $\tilde{x} = x_2$, $\tilde{y} = y_2$ и поиск заканчивается. Иначе перейти к шагу 2.

После k итераций общее число вычислений значения функции равно $N = 2k + 1$, а длина получившегося отрезка локализации будет:

$$L_N = \frac{b-a}{2^k} = \frac{b-a}{2^{\lfloor N/2 \rfloor}},$$

где $\lfloor z \rfloor$ – целая часть числа z .

Следовательно, достигнутая точность будет:

$$|x^* - \tilde{x}| \leq \varepsilon, \quad \varepsilon = 1/2 L_N.$$

1.1.3 Метод дихотомии

Это алгоритм блочного поиска для $n = 2$, т.е. когда в блоке два вычисления значения функции. Так как пассивная составляющая алгоритма, т.е. блок, содержит четное число точек, то оптимальный выбор точек x_{ij} , в которых необходимо вычислить значение функции, будет неравномерным, в отличие от предыдущих алгоритмов, где число точек в блоке было нечетным и, соответственно, расположение точек равномерным. Если блок содержит две точки, то они должны быть расположены как можно ближе к середине отрезка. Такое расположение точек позволяет получить наименьший отрезок неопределенностей.

Схема алгоритма.

Шаг_1. Задаются a, b, ε и δ – малое положительное число, значительно меньшее чем ε .

Шаг_2. Определяется середина отрезка:

$$x = (a + b) / 2.$$

Вычисляем значение функции в двух точках, близких к середине:

$$y_1 = f(x - \delta), \quad y_2 = f(x + \delta).$$

Шаг_3. Определяется следующий отрезок локализации, т.е. определяется какой из отрезков $[a, x + \delta]$ или $[x - \delta, b]$ содержит точное решение x^* . Если $y_1 \leq y_2$, то это отрезок $[a, x + \delta]$ и $b = x + \delta$, иначе это отрезок $[x - \delta, b]$ и $a = x - \delta$, т.е. выбранный отрезок локализации мы снова обозначили как $[a, b]$.

Шаг_4. Если $b - a \leq 2\varepsilon$, то

$$\tilde{x} = (a + b) / 2,$$

$$\tilde{y} = f(\tilde{x})$$

и поиск заканчивается. Иначе перейти к шагу 2.

После k итераций число вычислений функции – $N = 2k$, а длина отрезка локализации:

$$L_N < \frac{b - a}{2^{N/2}} + \delta.$$

Следовательно,

$$|x^* - \tilde{x}| < \frac{1}{2} L_N.$$

1.1.4 Метод золотого сечения

Для того чтобы уменьшить отрезок локализации $[a, b]$, необходимо вычислить значение целевой функции $f(x)$, по крайней мере, в двух точках на отрезке $[a, b]$ (рисунок 1.1). В результате этого отрезок локализации сузится до отрезка $[a, x_2]$ или $[x_1, b]$. Так как у нас нет никаких оснований предпочесть один из этих вариантов, то точки x_1 и x_2 должны быть симметричны относительно середины отрезка $[a, b]$. В этом случае длины отрезков $[a, x_2]$ и $[x_1, b]$ будут равны. Таким образом, остаётся вопрос: как выбрать точку x_1 .

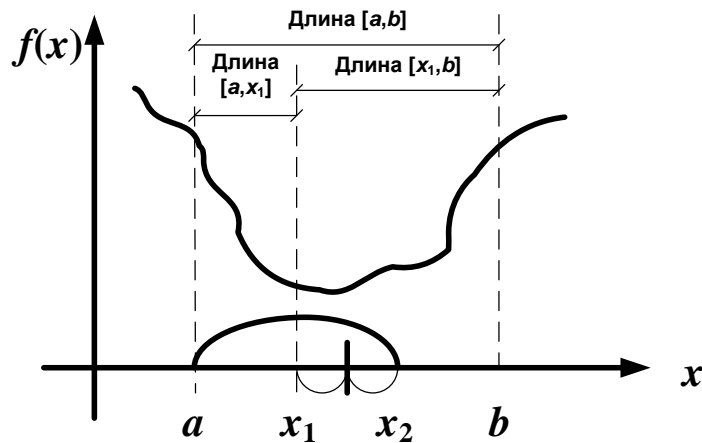


Рисунок 1.1 – Метод золотого сечения

В методе золотого сечения точка x_1 выбирается из соотношения:

$$\frac{\text{длина } [a, b]}{\text{длина } [x_1, b]} = \frac{\text{длина } [x_1, b]}{\text{длина } [a, x_1]} = \lambda = 1,618033989...,$$

т.е. точка x_1 делит отрезок $[a, b]$ по правилу «золотого сечения», где λ – есть «золотое отношение». Точка x_2 определяется как точка симметричная к x_1 относительно середины отрезка.

В результате вычисления значение целевой функции получается отрезок локализации $[a, x_2]$, содержащий точку x_1 , или отрезок локализации $[x_1, b]$, содержащий точку x_2 . Оказывается, что остающаяся точка на суженном отрезке локализации делит его вновь по правилу «золотого сечения». Следовательно, чтобы, в свою очередь, уменьшить новый отрезок локализации, требуется вычисления целевой

функции в точке, симметричной к оставшейся точке относительно середины этого нового отрезка.

В приводимой ниже схеме алгоритма остающиеся отрезки локализации переименовываются каждый раз как $[a, b]$, а точки, в которых проводятся вычисления целевой функции, обозначается через x_1 и x_2 , причём $x_1 < x_2$. Кроме того, $y_1 = f(x_1)$ и $y_2 = f(x_2)$.

Схема алгоритма

Шаг_1. Задаются a, b, ε и $\lambda = 1.618\dots$. Вычисляют:

$$x_1 = b - \frac{b-a}{\lambda}, \quad x_2 = a + \frac{b-a}{\lambda}, \quad y_1 = f(x_1), \quad y_2 = f(x_2).$$

Шаг_2.

а) Если $y_1 \leq y_2$, то полагают $b = x_2$, $x_2 = x_1$, $y_2 = y_1$ и вычисляют $x_1 = a + b - x_2$, $y_1 = f(x_1)$.

б) Если $y_1 > y_2$, то полагают $a = x_1$, $x_1 = x_2$, $y_1 = y_2$ и вычисляют $x_2 = a + b - x_1$, $y_2 = f(x_2)$.

Шаг_3. Если $b - a > \varepsilon$, то переходят к шагу 2. Иначе если $y_1 < y_2$, то полагают $\tilde{x} = x_1$ и $\tilde{y} = y_1$, если $y_1 \geq y_2$, то полагают $\tilde{x} = x_2$ и $\tilde{y} = y_2$.

После каждой итерации длина отрезка локализации уменьшается в λ раз. Так как первая итерация начинается после вычисления функции в двух точках, то после N вычислений длина отрезка локализации будет:

$$L_N = \frac{b-a}{\lambda^{N-1}}.$$

1.1.5 Метод чисел Фибоначчи

Этот метод применяется, когда число вычислений функции N заранее задано. Метод чисел Фибоначчи, также как и метод золотого сечения относится к симметричным методам, т.е. точки расположены симметрично относительно середины отрезка. Вот только выбор точки x_1 происходит на основе других соотношений. Для этого используются числа Фибоначчи: $F_0, F_1, F_2, F_3, \dots$, где $F_i = F_{i-2} + F_{i-1}$ ($i = 2, 3, \dots$) и $F_0 = F_1 = 1$.

Точка x_1 определяется из соотношения:

$$\frac{\text{длина } [a, x_1]}{\text{длина } [a, b]} = \frac{F_{N-2}}{F_N},$$

то есть

$$x_1 = a + (b - a) \frac{F_{N-2}}{F_N}.$$

Точка x_1 делит $[a, b]$ на две неравные части. Отношение малого отрезка к большему равно F_{N-2}/F_{N-1} . Точка x_2 определяется как точка, симметричная к x_1 относительно середины отрезка $[a, b]$:

$$x_2 = b - (b - a) \frac{F_{N-2}}{F_N} = a + (b - a) \frac{F_{N-1}}{F_N},$$

при этом будет выполняться условие $x_1 < x_2$.

В результате вычисления значение целевой функции в точках x_1 и x_2 получится отрезок локализации $[a, x_2]$, содержащий точку x_1 , или отрезок локализации $[x_1, b]$, содержащий точку x_2 . Остающаяся точка делит новый отрезок локализации на две неравные части в отношении F_{N-3}/F_{N-2} . То есть в методе Фибоначчи остающаяся точка делит отрезок на две неравные части в пропорциях, определяемых числами Фибоначчи. На k -ом шаге это отношение равно: $\frac{F_{N-K-2}}{F_{N-K}}$, а

длины отрезков равны: $\frac{F_{N-K-1}}{F_N}(b - a)$ и $\frac{F_{N-K}}{F_N}(b - a)$, соответственно.

Пример деления отрезка для случая, когда $a = 0$ и $b = 1$, представлен на рисунке 1.2. Если выполнить замену переменной

$$x' = a + x(b - a),$$

то полученные результаты будут перенесены на отрезок $[a, b]$.

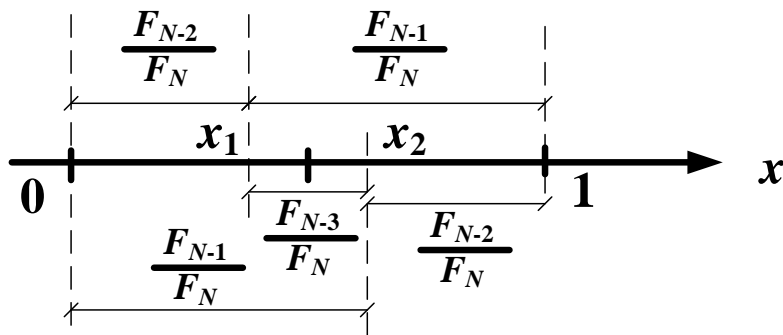


Рисунок 1.2 – Метод чисел Фибоначчи

Схема алгоритма

Шаг_1. Задаются a, b, N . Вычисляются числа Фибоначчи F_0, F_1, \dots, F_N . Определяется:

$$x_1 = a + (b - a)F_{N-2} / F_N,$$

$$x_2 = a + (b - a)F_{N-1} / F_N,$$

$$y_1 = f(x_1), \quad y_2 = f(x_2).$$

Шаг_2.

а) Если $y_1 \leq y_2$, то полагают $b = x_2, x_2 = x_1, y_2 = y_1$ и вычисляют $x_1 = a + b - x_2, y_1 = f(x_1)$.

б) Если $y_1 > y_2$, то полагают $a = x_1, x_1 = x_2, y_1 = y_2$ и вычисляют $x_2 = a + b - x_1, y_2 = f(x_2)$.

Повторить *Шаг_2* $N - 2$ раза.

Шаг_3. Если $y_1 < y_2$, то полагают $\tilde{x} = x_1$ и $\tilde{y} = y_1$. Если $y_1 \geq y_2$, то полагают $\tilde{x} = x_2$ и $\tilde{y} = y_2$.

Длина отрезка локализации в методе Фибоначчи:

$$L_N \approx (b - a) / F_N.$$

1.2 Линейный поиск с использованием производных

1.2.1 Метод касательных

Пусть функция $f(x)$ выпукла и дифференцируема на $[a, b]$. Идея метода состоит в следующем. Пусть $[a, b]$ – отрезок локализации и $f(a), f'(a), f(b), f'(b)$ – результаты вычислений в точках a и b . По этой информации строится аппроксимирующая функция, представляющую из себя кусочно-линейную функцию, состоящую из касательной

$$L_a(x) = f(a) + f'(a)(x - a)$$

к $f(x)$ в точке a и касательной

$$L_b(x) = f(b) + f'(b)(x - b)$$

к $f(x)$ в точке b .

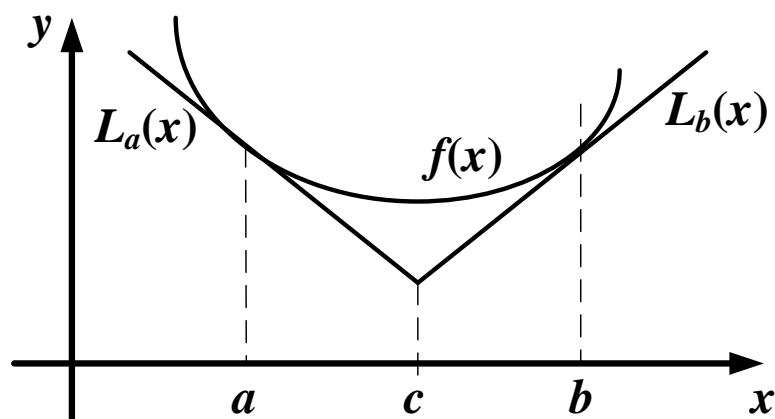


Рисунок 1.3 – Метод касательных

Полученная аппроксимирующая функция есть ломанная состоящая из прямой $L_a(x)$ на $[a, c]$ и $L_b(x)$ на $[c, b]$, где c – точка пересечения касательных. Легко заметить, что при $f(a) < 0$ и $f(b) > 0$ минимум аппроксимирующей функции достигается в точке c . Значение точки пересечения c можно вычислить по формуле

$$c = \frac{(bf'(b) - af'(a)) - (f(b) - f(a))}{f'(b) - f'(a)}.$$

В точке c производятся вычисления $f(c)$ и $f'(c)$. Если $f'(c) = 0$, то решением задачи будет $x^* = c$. Если же $f'(c) > 0$, то в качестве следующего отрезка локализации будет $[a, c]$. Если же $f'(c) < 0$, то – отрезок $[c, b]$. Процесс повторяется до тех пор, пока $f'(c) = 0$ или отрезок локализации не достигнет заданной точности.

Схема алгоритма

Шаг 1. Заданы a, b, ε . Вычислить:

$$y_1 = f(a), \quad y_2 = f(b), \quad z_1 = f'(a), \quad z_2 = f'(b).$$

Шаг 2. Если $b - a \leq 2\varepsilon$, то полагаем:

$$\tilde{x} = (a + b)/2, \quad \tilde{y} = f(\tilde{x}).$$

Поиск окончен.

Если $b - a > 2\varepsilon$, то вычислить:

$$c = \frac{(bz_2 - az_1) - (y_2 - y_1)}{z_2 - z_1}, \quad y = f(c), \quad z = f'(c).$$

Если $z = 0$, то полагаем $\tilde{x} = c$, $\tilde{y} = y$ и поиск окончен.

Если $z < 0$, то $a = c$, $y_1 = y$, $z_1 = z$. Если $z > 0$, то $b = c$, $y_2 = y$, $z_2 = z$. Повторить Шаг_2.

1.2.2 Метод кубической интерполяции

Суть алгоритма заключается в том, что на каждой итерации функция $f(x)$ аппроксимируется кубическим полиномом $f(x)$, точка минимума \tilde{x} которого берется в качестве текущего приближения к искомому минимуму x^* . Алгоритм предполагает вычисления в каждой очередной точке значений функции $f(x)$ и ее производной $f'(x)$.

Вначале необходимо задать параметры метода: ε – погрешность вычисления минимума, x_1 – начальную точку поиска, h_1 – направление поиска и установить начальный интервал $[a, b]$. Если $f'(x_1) > 0$, то изменить направление поиска: $h_1 = -h_1$. Вычислять $f'(x_k)$ в точках $x_{k+1} = x_k - h_k$ при $h_k = 2h_{k-1}$, $k = \overline{2, m-1}$ до тех пор, пока не продвигемся в точку x_m , такую, что $f'(x_m)f'(x_{m-1}) < 0$. Если $x_{m-1} < x_m$, то положить $a = x_{m-1}$, $b = x_m$, иначе $a = x_m$, $b = x_{m-1}$.

Схема алгоритма

Шаг_1. Вычислить параметр:

$$\gamma = \frac{z + w - f'(a)}{f'(b) - f'(a) + 2w},$$

где $z = f'(a) + f'(b) + 3 \frac{f(b) - f(a)}{b - a}$, $w = \sqrt{z^2 - f'(a)f'(b)}$ и при-

нять аппроксимирующий минимум:

–если $0 \leq \gamma \leq 1$, то $\tilde{x} = a + \gamma(b - a)$;

–если $\gamma < 0$, то $\tilde{x} = a$;

–если $\gamma > 1$, то $\tilde{x} = b$.

Шаг_2. Проверить критерий окончания поиска: $|f'(x^*)| < \varepsilon$, или $\tilde{x} = a$, или $\tilde{x} = b$. В противном случае вернуться к шагу 1, используя новый интервал:

–если $f'(x^*) > 0$, то $[a, x^*]$;

–если $f'(x^*) < 0$, то $[x^*, b]$.

2 ЗАДАНИЕ НА ЛАБОРАТОРНУЮ РАБОТУ

2.1. Изучить предлагаемые методы одномерной безусловной оптимизации.

2.2. Разработать программу, реализующую методы поиска (в соответствии с заданным вариантом), и найти точку минимума функции $f(x)$ на отрезке $[a, b]$.

Таблица 2.1 – Варианты заданий

№	Метод	Целевая функция	№	Метод	Целевая функция
1	$\bar{b}, \bar{v}, \bar{z}, \bar{d}$,3	$x^2 + 6 \cdot e^{0,15x}$	16	a, \bar{v}, \bar{z}, e ЖС	$1,4x + e^{ x-2 }$
2	a, \bar{v}, \bar{z}, e ЖС	$x^2 + 4 \cdot e^{-0,25x}$	17	a, \bar{b}, \bar{d}, e ,3	$x^2 + e^x$
3	a, \bar{b}, \bar{d}, e ,3	$x^4 + 0,4 \cdot \arctg 5x$	18	$a, \bar{b}, \bar{z}, \bar{d}$,ЖС	$ x + e^{10x}$
4	$a, \bar{b}, \bar{z}, \bar{d}$,ЖС	$x^4 - 1,5 \arctg x$	19	a, \bar{v}, \bar{z}, e 3	$10 \cos x + e^x$
5	a, \bar{v}, \bar{z}, e 3	$x^2 + 8 \cdot e^{0,55x}$	20	$\bar{b}, \bar{v}, \bar{d}, e$,ЖС	$-4x^2 + x^4$
6	$\bar{b}, \bar{v}, \bar{d}, e$,ЖС	$-4x + e^{ x-0,2 }$	21	a, \bar{b}, \bar{d}, e ,3	$x^2 + e^{ x-2 }$
7	$\bar{b}, \bar{v}, \bar{z}, \bar{d}$,3	$1,4x + e^{ x-2 }$	22	$a, \bar{b}, \bar{z}, \bar{d}$,ЖС	$x^4 + e^{x-2}$
8	a, \bar{v}, \bar{z}, e ЖС	$x^2 + e^x$	23	a, \bar{v}, \bar{z}, e 3	$x^2 + 6 \cdot e^{0,15x}$
9	a, \bar{b}, \bar{d}, e ,3	$ x + e^{10x}$	2	$\bar{b}, \bar{v}, \bar{d}, e$,ЖС	$x^2 + 4 \cdot e^{-0,25x}$
10	$a, \bar{b}, \bar{z}, \bar{d}$,ЖС	$10 \cos x + e^x$	25	$\bar{b}, \bar{v}, \bar{z}, \bar{d}$,3	$x^4 + 0,4 \cdot \arctg 5x$
11	a, \bar{v}, \bar{z}, e 3	$-4x^2 + x^4$	26	a, \bar{v}, \bar{z}, e ЖС	$x^4 - 1,5 \arctg x$
12	$\bar{b}, \bar{v}, \bar{d}, e$,ЖС	$x^2 + e^{ x-2 }$	27	a, \bar{b}, \bar{d}, e ,3	$x^2 + 8 \cdot e^{0,55x}$
13	$\bar{b}, \bar{v}, \bar{z}, \bar{d}$,3	$x^4 + e^{x-2}$	28	$a, \bar{b}, \bar{z}, \bar{d}$,ЖС	$x^2 + e^{ x-2 }$
14	a, \bar{v}, \bar{z}, e ЖС	$ x^3 + e^x$	29	a, \bar{v}, \bar{z}, e 3	$x^4 + e^{x-2}$

15	a, \bar{b}, ∂, e ,з	$7 \cos x + e^{x+3}$	30	$\bar{b}, \bar{v}, \partial, e$,ж	$ x^3 + e^x$
----	---------------------------------	----------------------	----	---------------------------------------	---------------

Методы одномерной безусловной оптимизации:

- а) пассивный оптимальный алгоритм;*
- б) алгоритм блочного равномерного поиска;*
- в) алгоритм деления интервала пополам;*
- г) метод дихотомии;*
- д) метод золотого сечения;*
- е) метод Фибоначчи;*
- ж) метод касательных;*
- з) метод кубической интерполяции.*

3 ТРЕБОВАНИЕ К ОТЧЕТУ

В отчете должны быть отображены следующие пункты:

1. Задание.
2. Краткие теоретические сведения.
3. График функции в *MathCAD*.
4. Схемы алгоритмов.
5. Листинги основных частей программы.
6. Таблица результатов сравнения рассмотренных методов.
7. Заключение по результатам сравнения методов.

4 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое экстремум функции.
2. Как аналитически найти экстремум функции.
3. Пассивный оптимальный алгоритм поиска минимума.
4. Блочный равномерный поиск.
5. Алгоритм деления интервала пополам.
6. Метод дихотомии.
7. Метод золотого сечения.
8. Метод Фибоначчи.
9. Метод касательных.
10. Метод кубической интерполяции.