

НЕЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

(функция **fmincon**)

Функция **fmincon**, которая входит в пакет **Toolbox Optimization**, позволяет находить минимум для скалярной функции нескольких аргументов при заданном начальном приближении и наличии линейных и нелинейных ограничений (задачи нелинейного программирования).

Общая постановка задачи:

найти $\min_{\mathbf{x}} (f(\mathbf{x}))$ среди всех векторов \mathbf{x} , удовлетворяющих системе неравенств и равенств:

$$c(\mathbf{x}) \leq 0; \quad c_{eq}(\mathbf{x}) = 0; \quad \mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}; \quad \mathbf{A}_{eq} \cdot \mathbf{x} = \mathbf{b}_{eq}; \quad \mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}.$$

Обращение к функции **fmincon** выглядит следующим образом:

x=fmincon (fun , x0 , A , b , Aeq , beq , lb , ub , nonlcon , options , P1 , P2 , ...) .

Если выходных значений несколько:

[x,fval]=fmincon (...) – возвращает не только оптимальное значение векторного аргумента \mathbf{x} , но и значение целевой функции в точке минимума $fval = \min_{\mathbf{x}} (f(\mathbf{x}))$;

[x,fval, exitflag]=fmincon (...) – то же, что и предыдущая функция, но возвращается ещё информация о характере завершения вычислений:

exitflag > 0 – результат найден с требуемой точностью;
exitflag = 0 – достигнуто максимальное число итераций в процессе решения;
exitflag < 0 – решение не найдено.

Аргументы функции **fmincon()**

fun – подлежащая минимизации функция $f(\mathbf{x})$, которая может зависеть от нескольких параметров. Значения этих параметров, в случае их наличия, передаются в последних аргументах **P1, P2, ...**. Данная функция может задаваться с помощью описателя функций

x=fmincon (@myfun , x0 , A , b) ,

где **myfun** – функция MatLab в виде m-файла:

function F = myfun (x)

F = ... % Вычисление функции в точке \mathbf{x} .

x0 – начальное приближение к решению;

A – матрица линейных ограничений в виде неравенства: $\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}$;

b – вектор линейных ограничений в виде неравенства;

Aeq – матрица линейных ограничений в виде равенства: $\mathbf{A}_{eq} \cdot \mathbf{x} = \mathbf{b}_{eq}$;

beq – вектор линейных ограничений в виде равенства;

lb – вектор нижней границы двусторонних покомпонентных ограничений
 $\mathbf{lb} \leq \mathbf{x} \leq \mathbf{ub}$;

ub – вектор верхней границы двусторонних покомпонентных ограничений;

*Неиспользуемые векторы и матрицы ограничений заменяются в списке входных аргументов **fmincon** квадратными скобками (пустым массивом). Последние идущие подряд аргументы могут быть опущены.*

nonlcon – имя файл-функции, в которой запрограммированы нелинейные ограничения (равенства и неравенства). Входным аргументом **nonlcon** является вектор **x**, соответствующий искомому вектору **x**, а двумя выходными аргументами – вектора **c** и **ceq** левых частей нелинейных ограничений $c(\mathbf{x}) \leq 0$ и $c_{eq}(\mathbf{x}) = 0$:

```
function [c,ceq]=mycon(x)
c =... % Вычисл. левых частей нелин. неравенств
ceq =... % Вычисл. левых частей нелин. равенств.
```

options – данный аргумент предназначен для установки параметров, управляющих процессом минимизации. Данные параметры можно изменять, используя функцию **optimset**:

```
options = optimset('Display', 'iter').
```

Задачи оптимизации условно разделены в Toolbox Optimization на два класса **Medium-Scale** (средние, среднемасштабные) и **Large-Scale** (большие, крупномасштабные), в зависимости от размерности задачи, т.е. числа переменных. Для решения каждого из классов задач реализованы соответствующие численные методы. Часть параметров, задаваемых аргументом **options**, используется во всех алгоритмах оптимизации. Некоторые используются только для крупномасштабного алгоритма, другие применимы только для среднемасштабного алгоритма.

Для использования крупномасштабного алгоритма в **fun** необходимо задать градиент минимизируемой функции в точке **x**. При этом данный алгоритм наиболее эффективен в том случае, когда также рассчитывается матрица вторых производных, т.е. матрица Гессе.

Применение алгоритма среднемасштабной оптимизации не требует вычисления в **fun** градиента минимизируемой функции. При этом для поиска минимума используется метод Последовательного Квадратичного Программирования (SQP).

Функция **optimset('fmincon')** выводит в командное окно структуру с информацией о текущих параметрах вычислительного алгоритма. Обращение к

данной функции без входных аргументов позволяет отобразить в командном окне все возможные значения каждого из параметров.

Рассмотрим назначение основных параметров оптимизации, учитывая последующее использование среднемасштабного алгоритма.

Параметры, используемые как для среднемасштабного, так и для крупномасштабного алгоритмов (Medium-Scale, Large-Scale Algorithms):

LargeScale – устанавливает преимущественное право использования крупномасштабного алгоритма (алгоритма большой размерности). Может принимать значения **'off'** (по умолчанию) и **'on'**. В первом случае используется алгоритм средней размерности, во втором – алгоритм большой размерности;

Diagnostics – проводится печать диагностической информации о минимизируемой функции;

Display – уровень отображения:

'off' – вывод информации отсутствует,

'iter' – вывод информации о поиске решения на каждой итерации,

'final' – (по умолчанию) вывод только итоговой информации;

GradObj – использование градиента для целевой функции, определяемого пользователем (возможные значения – **'off'** и **'on'**);

MaxFunEvals – максимально-допустимое число вычислений функции;

MaxIter – максимальное допустимое число итераций;

TolFun – допуск останова вычислений по величине изменений функции;

TolCon – допуск останова вычислений при нарушении ограничений;

TolX – конечное допустимое отклонение по значению **x**;

DerivativeCheck – задает проверку соответствия производных, определенных пользователем, их вычисленным оценкам в виде первых конечных разностей
и др.

Параметры, используемые только для среднемасштабного алгоритма:

DiffMaxChange – максимальные значения изменений переменных при определении первых конечных разностей;

DiffMinChange – минимальные значения изменений переменных при определении первых конечных разностей;

LineSearchType – задание вида алгоритма одномерной оптимизации
и др.

Параметры, используемые только для крупномасштабного алгоритма:

Hessian, **HessMult**, **HessPattern**, **MaxPCGIter**, **PrecondBandWidth**, **TolPCG**, **JacobMult**, **JacobPattern**.