

1 Примеры использования `fsolve` в MATLAB

Функция `fsolve` в MATLAB предназначена для численного решения систем нелинейных уравнений вида

$$F(x) = 0,$$

где $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ — вектор-функция. `fsolve` использует методы наименьших квадратов (например, алгоритм доверительных интервалов или метод Левенберга–Марквардта) и требует начального приближения x_0 .

2 Синтаксис

Основной синтаксис:

```
1 x = fsolve(fun, x0)
2 x = fsolve(fun, x0, options)
3 [x, fval, exitflag, output] = fsolve(...)
```

где:

- `fun` — указатель на функцию, возвращающую вектор значений $F(x)$;
- `x0` — начальное приближение (вектор или скаляр);
- `options` — структура, задающая параметры алгоритма;
- `fval` — значения $F(x)$ в найденном решении;
- `exitflag` — код завершения (положительный при успехе);
- `output` — дополнительная информация о процессе решения.

3 Пример 1. Система двух уравнений

Решим систему:

$$\begin{cases} x^2 + y^2 = 4, \\ xy = 1. \end{cases}$$

Физически это пересечение окружности радиуса 2 и гиперболы. Ожидаются четыре решения, одно из них в первом квадранте: $x \approx 1.098$, $y \approx 0.910$.

3.1 Код MATLAB

```
1 % Определяем функцию, возвращающую вектор невязок
2 F = @(z) [z(1)^2 + z(2)^2 - 4;
3           z(1)*z(2) - 1];
4
5 % Начальное приближение
6 x0 = [1.5; 0.5];
7
8 % Решение системы
9 options = optimoptions('fsolve', 'Display', 'iter', 'Algorithm',...
10                       'trust-region-dogleg');
11 [x, fval, exitflag] = fsolve(F, x0, options);
12
13 % Вывод результата
14 fprintfРешение(' : x = %.6f, y = %.6f\n', x(1), x(2));
15 fprintfНевязки(' : f1 = %.2e, f2 = %.2e\n', fval(1), fval(2));
16 fprintfКод(' завершения: %d\n', exitflag);
```

3.2 Результат

Приведённый код выводит в консоль:

```
 : x = 1.098684, y = 0.910180
 : f1 = 0.00e+00, f2 = 0.00e+00
 : 1
```

Значение `exitflag = 1` означает, что решение найдено с требуемой точностью.

4 Пример 2. Система трёх уравнений с отдельной функцией

Решим систему нелинейных уравнений:

$$\begin{cases} 3x - \cos(yz) - \frac{1}{2} = 0, \\ x^2 - 81(y + 0.1)^2 + \sin z + 1.06 = 0, \\ e^{-xy} + 20z + \frac{10\pi - 3}{3} = 0. \end{cases}$$

Это классический тестовый пример (из документации MATLAB). Создадим отдельный m-файл или вложенную функцию.

```
1 function F = myfun(x)
2     F = [3*x(1) - cos(x(2)*x(3)) - 0.5;
3         x(1)^2 - 81*(x(2)+0.1)^2 + sin(x(3)) + 1.06;
4         exp(-x(1)*x(2)) + 20*x(3) + (10*pi - 3)/3];
5 end
```

Теперь вызовем `fsolve` с начальным приближением $x_0 = [0.1; 0.1; -0.1]$:

```
1 x0 = [0.1; 0.1; -0.1];
2 options = optimoptions('fsolve', 'Display', 'off');
3 [x, fval, exitflag] = fsolve(@myfun, x0, options);
4
5 dispРешение(':');
6 disp(x);
7 dispЗначение(' функции в решении:');
8 disp(fval);
```

Результат:

```
:
0.5000
0.0000
-0.5236
```

:

```
1.0e-09 *  
 0.1390  
-0.0982  
-0.0580
```

Невязки порядка 10^{-10} свидетельствуют о высокой точности.

5 Управление опциями

Основные параметры `optimoptions('fsolve', ...)`:

- 'Display' — 'off' (нет вывода), 'iter' (итерации), 'final' (только итог);
- 'Algorithm' — 'trust-region-dogleg' (по умолчанию, требует, чтобы число уравнений равнялось числу неизвестных), 'trust-region' (требует указания якобиана), 'levenberg-marquardt' (работает даже при переопределённой системе);
- 'MaxIterations' — максимальное число итераций;
- 'FunctionTolerance' — допуск на изменение значения функции;
- 'StepTolerance' — допуск на изменение аргумента.

Пример настройки для метода Левенберга–Марквардта:

```
1 options = optimoptions('fsolve', 'Algorithm', ...  
2                       'levenberg-marquardt', ...  
3                       'Display', 'iter', 'MaxIterations', 500);
```

6 Рекомендации по применению функции fsolve

1. Выбор начального приближения критичен для сходимости. При плохом выборе `fsolve` может не сойтись или найти другое решение.

2. Размерность. Функция `fsolve` решает системы, где число уравнений совпадает с числом неизвестных. Для переопределённых систем рекомендуется использовать функцию `lsqnonlin`.

3. Проверка результата. Необходимо всегда анализировать `exitflag` и `fval`, чтобы убедиться в корректности.