Johns Hopkins Engineering

Applied Machine Learning for Mechanical Engineers

Optimization, Part 2, C

JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

Introduction to Nonlinear Optimization Programming Packages

- By the end of this lecture you will be able to:
  - Address nonlinear optimization problems such to be compatible with Python and MATLAB optimization packages
  - Describe “scipy.optimize.minimize” package in Python
  - Describe ‘fmincon” package in MATLAB

Introduction to Nonlinear Optimization Programming Packages

- Formal nonlinear optimization problem with one objective function

$$\begin{gathered} \mathit{minimize}\ f(\boldsymbol{x}) \\ \mathit{subject\ to}\ \begin{cases} g_j(\boldsymbol{x}) \le 0 & j \in \{1, 2, \dots, J\} \\ h_k(\boldsymbol{x}) = 0 & k \in \{1, 2, \dots, K\} \end{cases} \end{gathered} \qquad (2-3)$$

where $\boldsymbol{x} = [x_1, x_2, \dots, x_N]$ include the optimization variables (solution), $f(\boldsymbol{x})$, $g_j(\boldsymbol{x})$, and $h_k(\boldsymbol{x})$ are a linear/nonlinear objective function, linear/nonlinear inequality constraints, and linear/nonlinear equality constraints, respectively, and the rest is similar to Eq. 1-1.

Introduction to Nonlinear Optimization Programming Packages

- Programming packages for nonlinear programming

$$minimize f(\boldsymbol{x})$$
$$subject\ to \begin{cases} g_j(\boldsymbol{x}) \ \leq 0 \\ h_k(\boldsymbol{x}) = 0 \\ \boldsymbol{G}.\boldsymbol{x} \leq \boldsymbol{A} \\ \boldsymbol{H}.\boldsymbol{x} = \boldsymbol{B} \\ \boldsymbol{L} \leq \boldsymbol{x} \leq \boldsymbol{U} \end{cases} \qquad (2-4)$$

where $\boldsymbol{x} = [x_1, x_2, \dots, x_N]$ include the optimization variables (solution), $f(\boldsymbol{x})$, $g_j(\boldsymbol{x})$, and $h_k(\boldsymbol{x})$ are a linear/nonlinear objective function, nonlinear (only) inequality constraints, and nonlinear equality constraints (only), respectively, and the rest is similar to Eq. 2-2.

Introduction to Nonlinear Optimization Programming Packages

- Programming packages for nonlinear programming
  - Different annotations

$$\min_{x} f(x) \text{ such that } \begin{cases} c(x) \leq 0 \\ ceq(x) = 0 \\ A \cdot x \leq b \\ Aeq \cdot x = beq \\ lb \leq x \leq ub, \end{cases}$$

Figure 2-2 Annotations used to address nonlinear optimization problems at https://www.mathworks.com/help/optim/ug/fmincon.html

Introduction to Nonlinear Optimization Programming Packages

- Programming packages for nonlinear programming
  - Python: “scipy.optimize.minimize” at

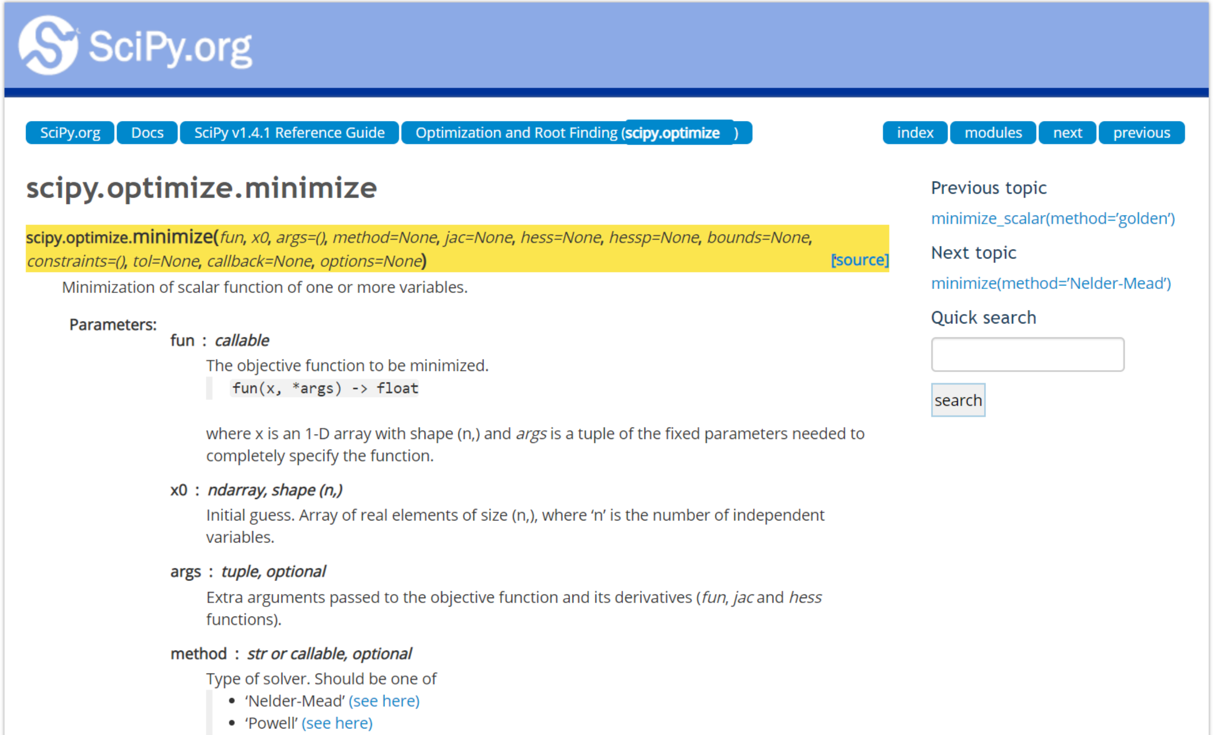
    https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize

  - MATLAB: “fmincon” at

    https://www.mathworks.com/help/optim/ug/fmincon.html

Introduction to Nonlinear Optimization Programming Packages

- Python: “scipy.optimize.minimize”



SciPy.org
SciPy.org
Docs
SciPy v1.4.1 Reference Guide
Optimization and Root Finding (scipy.optimize)
index
modules
next
previous

scipy.optimize.minimize

scipy.optimize.minimize(*fun, x0, args=(), method=None, jac=None, hess=None, hessp=None, bounds=None, constraints=(), tol=None, callback=None, options=None*) [source]

Minimization of scalar function of one or more variables.

Parameters:

fun : *callable*

The objective function to be minimized.

```
fun(x, *args) -> float
```

where x is an 1-D array with shape (n,) and *args* is a tuple of the fixed parameters needed to completely specify the function.

x0 : *ndarray, shape (n,)*

Initial guess. Array of real elements of size (n,), where ‘n’ is the number of independent variables.

args : *tuple, optional*

Extra arguments passed to the objective function and its derivatives (*fun, jac* and *hess* functions).

method : *str or callable, optional*

Type of solver. Should be one of

- ‘Nelder-Mead’ (see here)
- ‘Powell’ (see here)

Previous topic

minimize_scalar(method=’golden’)

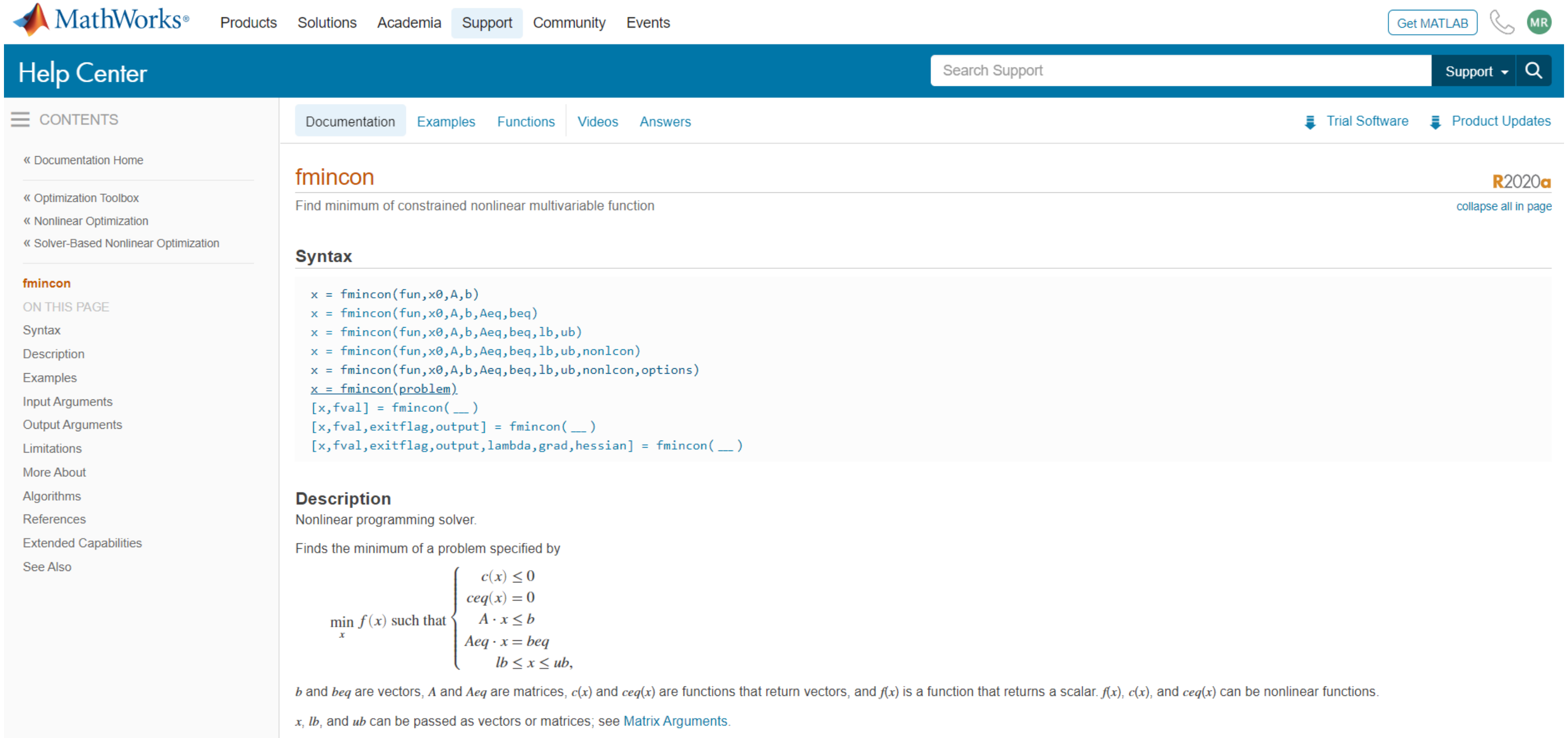
Next topic

minimize(method=’Nelder-Mead’)

Quick search

Introduction to Nonlinear Optimization Programming Packages

- MATLAB: “fmincon”



fmincon

Find minimum of constrained nonlinear multivariable function

Syntax

```
x = fmincon(fun,x0,A,b)
x = fmincon(fun,x0,A,b,Aeq,beq)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon)
x = fmincon(fun,x0,A,b,Aeq,beq,lb,ub,nonlcon,options)
x = fmincon(problem)
[x,fval] = fmincon(___)
[x,fval,exitflag,output] = fmincon(___)
[x,fval,exitflag,output,lambda,grad,hessian] = fmincon(___)
```

Description

Nonlinear programming solver.

Finds the minimum of a problem specified by

$$\min_{x} f(x) \text{ such that } \begin{cases} c(x) \le 0 \\ ceq(x) = 0 \\ A \cdot x \le b \\ Aeq \cdot x = beq \\ lb \le x \le ub, \end{cases}$$

b and beq are vectors, A and Aeq are matrices, $c(x)$ and $ceq(x)$ are functions that return vectors, and $f(x)$ is a function that returns a scalar. $f(x)$, $c(x)$, and $ceq(x)$ can be nonlinear functions.

x, lb, and ub can be passed as vectors or matrices; see Matrix Arguments.

Introduction to Nonlinear Optimization Programming Packages

- In this lecture, you learned about:

  - General formulations of nonlinear optimization problems in Python and MATLAB optimization packages

  - “scipy.optimize.minimize” programming package in Python

  - ‘fmincon” programming package in MATLAB

- In the next lecture, we will practice these programming packages in Python and MATLAB

JOHNS HOPKINS

WHITING SCHOOL
of ENGINEERING