

# 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

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \begin{cases} g_j(\mathbf{x}) \leq 0 & j \in \{1, 2, \dots, J\} \\ h_k(\mathbf{x}) = 0 & k \in \{1, 2, \dots, K\} \end{cases} \end{array} \quad (2-3)$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_N]$  include the optimization variables (solution),  $f(\mathbf{x})$ ,  $g_j(\mathbf{x})$ , and  $h_k(\mathbf{x})$  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

$$\begin{array}{ll} \text{minimize} & f(\mathbf{x}) \\ \text{subject to} & \left\{ \begin{array}{l} g_j(\mathbf{x}) \leq 0 \\ h_k(\mathbf{x}) = 0 \\ \mathbf{G} \cdot \mathbf{x} \leq \mathbf{A} \\ \mathbf{H} \cdot \mathbf{x} = \mathbf{B} \\ \mathbf{L} \leq \mathbf{x} \leq \mathbf{U} \end{array} \right. \end{array} \quad (2-4)$$

where  $\mathbf{x} = [x_1, x_2, \dots, x_N]$  include the optimization variables (solution),  $f(\mathbf{x})$ ,  $g_j(\mathbf{x})$ , and  $h_k(\mathbf{x})$  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   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”

The screenshot shows the MATLAB Help Center interface. At the top, there's a navigation bar with links to Products, Solutions, Academia, Support, Community, and Events. A search bar is also present. The main header is "Help Center". On the left, a "CONTENTS" sidebar lists various topics, with "fmincon" highlighted under "Nonlinear Optimization". The main content area is titled "fmincon" and describes it as a function to find the minimum of a constrained nonlinear multivariable function. It includes a "Syntax" section with several code snippets showing different ways to call the function, a "Description" section explaining the problem it solves, and a mathematical formulation of the optimization problem.

MathWorks® Products Solutions Academia Support Community Events

Get MATLAB

Help Center

Search Support

Support

Documentation Examples Functions Videos Answers

Trial Software Product Updates

### fmincon

Find minimum of constrained nonlinear multivariable function

R2020a

collapse all in page

#### 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

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

$b$  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