Sessions 06-08

Scientific Tools

(in Python and MATLAB)

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

1.Equation Solving
2.Optimization
3.Differentiation
4.Integration
5.Interpolation
6.Linear Algebra (Matrix Operations, Eigenvalues and Eigenvectors, Singular Value Decompositions, etc.)
7.Regression
8.Statistical Tests
9.Random Variables and Distributions

(Nonlinear) Equation Solving

$$f(x) = 0$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Python `scipy.optimize.fsolve`

https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.optimize.fsolve.html

001 010

scipy.optimize.fsolve

scipy.optimize.fsolve(*func, x0, args=(), fprime=None, full_output=0, col_deriv=0, xtol=1.49012e-08, maxfev=0, band=None, epsfcn=None, factor=100, diag=None***)** [source]

Find the roots of a function.

Return the roots of the (non-linear) equations defined by `func(x) = 0` given a starting estimate.

`func` and `x0` are required arguments.

How to use `args`? (No example is given??!!!)

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

How to use `args`?

010

scipy.optimize.fsolve

scipy.optimize.fsolve(*func*, *x0*, *args*=(), *fprime*=None, *full_output*=0, *col_deriv*=0, *xtol*=1.49012e-08, *maxfev*=0, *band*=None, *epsfcn*=None, *factor*=100, *diag*=None) [source]

Parameters:

func : *callable* `f(x, *args)`
A function that takes at least one (possibly vector) argument.

x0 : *ndarray*
The starting estimate for the roots of `func(x) = 0`.

args : *tuple, optional*
Any extra arguments to *func*.

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

MATLAB `fzero`

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

fzero

Root of nonlinear function

Syntax

```
x = fzero(fun,x0)
x = fzero(fun,x0,options)

x = fzero(problem)

[x,fval,exitflag,output] = fzero( __ )
```

002

x0

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

fzero(fun, x0)

x0 — Initial value

scalar | 2-element vector

Initial value, specified as a real scalar or a 2-element real vector.

- Scalar — `fzero` begins at `x0` and tries to locate a point `x1` where `fun(x1)` has the opposite sign of `fun(x0)`. Then `fzero` iteratively shrinks the interval where `fun` changes sign to reach a solution.
- 2-element vector — `fzero` checks that `fun(x0(1))` and `fun(x0(2))` have opposite signs, and errors if they do not. It then iteratively shrinks the interval where `fun` changes sign to reach a solution. An interval `x0` must be finite; it cannot contain `±Inf`.

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

MATLAB `fsolve`

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

003

fsolve

R2018a

Solve system of nonlinear equations

collapse all in page

Nonlinear system solver

Solves a problem specified by

$$F(x) = 0$$

for x, where $F(x)$ is a function that returns a vector value.

x is a vector or a matrix; see Matrix Arguments.

Syntax

```
x = fsolve(fun,x0)
x = fsolve(fun,x0,options)
x = fsolve(problem)
[x,fval] = fsolve(__)
[x,fval,exitflag,output] = fsolve(__)
[x,fval,exitflag,output,jacobian] = fsolve(__)
```

This x0 is not that x0.

x0 — Initial point
real vector | real array

Initial point, specified as a real vector or real array. `fsolve` uses the number of elements in and size of `x0` to determine the number and size of variables that `fun` accepts.

Example: `x0 = [1,2,3,4]`

Data Types: `double`

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 1: Solve $x^2 = 2$

1. Write the equation into $f(x) = 0$ format

$$f(x) = x^2 - 2 = 0$$

004

2. Define function $f(x)$

(in-class exercise: Python, MATLAB)

3. Apply the library function

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Review: Define a function (1)

$$f(x) = x^2 - 2 = 0$$

004

```
f01=lambda x: x**2-2
```

```
f01=@(x) x^2-2;
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Review: Define a function (2)

$$f(x) = x^2 - 2 = 0$$

004

```
def f02(x):
    return x**2-2
```

```
function y=f02(x)
    y=x^2-2;
end
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

MATLAB function functions

https://www.mathworks.com/help/matlab/matlab_prog/creating-a-function-handle.html

When passing a function to another function as an input, MATLAB uses the **function handle**.

```
clear;
clc;
f01=@(x) x^2-2;
x=fzero(f01, 1)
[x, fval, exitflag, output]=fzero(f01, 1)
```

004

```
clear;
clc;
x=fzero(@f02, 1)
[x, fval, exitflag, output]=fzero(@f02, 1)
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

fzero

002

Example 1a: Use `fzero` to solve $x^2 = 2$ for x.

```
f01=@(x) x^2-2;
x=fzero(f01, 1)
[x, fval, exitflag, output]=fzero(f01, 1)
```

Alternatively, we can use

```
clear;
clc;
x=fzero(@f02, 1)
[x, fval, exitflag, output]=fzero(@f02, 1)
```

```
x = 1.4142
x = 1.4142
fval = 4.4409e-16
exitflag = 1
output =
    intervaliterations: 9
            iterations: 6
             funcCount: 25
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0.547452, 1.45255]'
```

```
x = 1.4142
x = 1.4142
fval = 4.4409e-16
exitflag = 1
output =
    intervaliterations: 9
            iterations: 6
             funcCount: 25
             algorithm: 'bisection, interpolation'
               message: 'Zero found in the interval [0.547452, 1.45255]'
```

(Dr. Z: How about using **fsolve**? Note that in MATLAB, a 1x1 vector/matrix is a scalar.)

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666 Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

fsolve

Example 1b: Use `fsolve` to solve $x^2 = 2$ for x.

```
f01=@(x) x^2-2;
x=fsolve(f01, 1)
[x, fval, exitflag, output]=fsolve(f01, 1)
```

Alternatively, we can use

```
clear;
clc;
x=fsolve(@f02, 1)
[x, fval, exitflag, output]=fsolve(@f02, 1)
```

```
Equation solved.

fsolve completed because the vector of function values is
as measured by the default value of the function toleranc
the problem appears regular as measured by the gradient.

<stopping criteria details>
x = 1.4142
Equation solved.

fsolve completed because the vector of function values is
as measured by the default value of the function toleranc
the problem appears regular as measured by the gradient.

<stopping criteria details>
x = 1.4142
fval = 4.5040e-12
exitflag = 1
output =
       iterations: 4
        funcCount: 10
        algorithm: 'trust-region-dogleg'
    firstorderopt: 1.2739e-11
          message: 'Equation solved.…'
Equation solved.

fsolve completed because the vector of function values is near zero
as measured by the default value of the function tolerance, and
the problem appears regular as measured by the gradient.

<stopping criteria details>
```

003

004

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Python function functions

fsolve

```
from scipy.optimize import fsolve
```

001

004

Example 1: Solve

$$x^2 = 2$$

```
f01=lambda x: x**2-2
def f02(x):
    return x**2-2

x=fsolve(f01, 1)
print(x)

x=fsolve(f02, 1)
print(x)
```

```
[1.41421356]
[1.41421356]
```

fsolve returns a list.

005

(Dr. Z: In Python, a list of 1 element won't reduce to a number.)

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Define myfzero

```
def myfzero
```

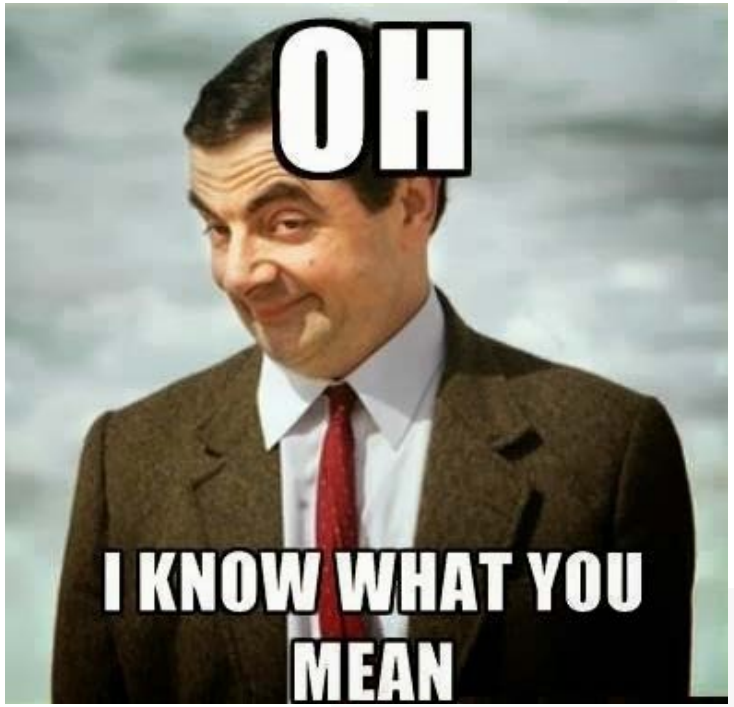
```
f01=lambda x: x**2-2
def f02(x):
    return x**2-2

x=myfzero(f01, 1)
print(x)

x=myfzero(f02, 1)
print(x)
```

```
1.4142135623730947
1.4142135623730947
```

006

05 min

myfzero



myfzero returns a number.

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

$$f(x) = x^2 - 2 = 0$$

```
# How to obtain the negative root of
#              x**2-2=0
# Method 1
x=fsolve(f02, -1)[0]
print(x)
# Method 2
from scipy.optimize import bisect
x=bisect(f02, -10, 0)
print(x)

-1.4142135623730947
-1.414213562374016
```

007

Q: How to obtain the negative root?

How to obtain the negative root?

```
% Method 1a
x=fzero(@f02, -1)

% Method 1b
x=fsolve(@f02, -1)

% Method 2
x=fzero(@f02, [-10, 0])
```

```
exitflag = 1
output =
        iterat
         func
         algo
    firstorde
          mes
x = -1.4142
Equation solv

fsolve comple
as measured b
the problem a

<stopping cri
x = -1.4142
x = -1.4142
```

bisect

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

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

scipy.optimize.bisect

scipy.optimize.**bisect**(*f*, *a*, *b*, *args=()*, *xtol=2e-12*, *rtol=8.881784197001252e-16*, *maxiter=100*, *full_output=False*, *disp=True*) [source]

Find root of a function within an interval.

Basic bisection routine to find a zero of the function *f* between the arguments *a* and *b*. *f(a)* and *f(b)* cannot have the same signs. Slow but sure.

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and
Computational
Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative
Finance

MATLAB functions that return a function/functions

```
function f=flist(x,y)
    fun_list={@f1, @f2};
    f=fun_list{mod(x+y,2)+1};
end

function y=f1(x)
    y=2*x;
end

function y=f2(x)
    y=x^2;
end
```

flist.m

不知道你说啥

```
fun=@(a,b) @(x) a*x+b;
```

Functions that return a function (1)

```
f=flist(1,4);
f(3)
```

ans = 9

Functions that return a function (2)

```
fun=@(a,b) @(x) a*x+b;
f=fun(2,3);
f(4)
```

ans = 11

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Python functions that return a function/functions

Functions that return a function/functions (1)

```
def f1(x):
    return 2*x
def f2(x):
    return x**2
def flist(x, y):
    fun_list=[f1, f2]
    return fun_list[(x+y)%2]
f=flist(1,4)
print(f(3))

fun=lambda a, b: lambda x: a*x+b
f=fun(2,3)
print(f(4))
```

```
9
11
```

008

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Python closures (revisited)

Functions that return a function/functions (2)

```
def flist2():
    return [lambda x: i*x for i in [3, 5]]
f3,f5=flist2()
print(f3(5), f5(5))
```

25 25

```
#What is the output?
i=3
f=lambda x: i*x
i=5
print(f(5))
i=7
print(f(5))
```

f(x)=i*x

25
35

009 008

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

How about MATLAB?

```
function f=flist2()
    fun_list=???????????
    for i=[3, 5]
        fun_list=?fun_list,??????????????????;
    end
    f=fun_list;
end
```

How to create a list of function handles?

Do Not Test

Functions that return a function (2)

```
f=flist2();
[f3,f5]=f{:};
[f3(5), f5(5)]
```

```
ans = 1x2 double
    15    25
```

```
i=3;
f=@(x) i*x;
i=5;
disp(f(5));
i=7;
disp(f(5));
```

```
15
15
```

f(x)=3*x

009 008

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 2: Given $f(x,a,b) = ax^3 + b$, solve

$$f(x,2,3) = 2x^3 + 3 = 0$$

010

```
def fxab(x,a,b):
    return a*x**3+b
```

```
function y=fxab(x,a,b)
    y=a*x^3+b;
end
```

How to use existing multi-variable functions in equation solving?

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

How to use existing multi-variable functions in equation solving?

How to define a one-variable function from an existing multi-vairable function?

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 2: Given `fxab(x,a,b)`, solve `fxab(x,2,3)=0`.

```
import numpy as np
def fxab(x, a, b):
    return a*x**3+b
x=fsolve(fxab, 1, args=(2,3))[0]
x
```

010

011

```
-1.144714242533323
```

```
(-3/2)**(1/3)
```

```
(0.572357121276666+0.991351614125047j)
```

```
-(3/2)**(1/3)
```

```
-1.144714242533319
```

```
>> (-3/2)^(1/3)
ans =
    0.5724 + 0.9914i
>> nthroot(-3/2,3)
ans =
   -1.1447
>> -(3/2)^(1/3)
ans =
   -1.1447
```

Python does not have **nthroot**.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

mynthroot


```
def mynthroot(x, n):
```

??????????????

```
mynthroot(-3/2, 3)
```

```
-1.1447142425533319
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

@(x) fxab(x, a, b)

Example 2: Given $f(x,a,b) = ax^3 + b$, solve $f(x,2,3) = 0$.

```
a=2;
b=3;
x=fzero(@(x) fxab(x,a,b), 1)
```

```
x = -1.1447
```

lambda x: fxab(x, a, b)

010

```
a=2
b=3
x=fsolve(lambda x: fxab(x,a,b), 1)[0]
x
```

```
-1.1447142425533323
```

(Dr. Z: What if it is $f(a,x,b)$?

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 3: Implied Volatility

$$c = BS(S, K, r, \sigma, T) = S \cdot e^{-qT} \cdot \Phi(d_1) - K \cdot e^{-rT} \cdot \Phi(d_2)$$

012

013

where

$$d_1 = \frac{\ln\left(\frac{S}{K}\right) + \left(r - q + \frac{\sigma^2}{2}\right)T}{\sigma\sqrt{T}}, \qquad d_2 = d_1 - \sigma\sqrt{T}$$

```
S=490, K=470, r=0.033, q=0, T=0.08, c=24.5941 ⇨ sigma
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

User-Defined Function for the Black-Scholes Formula

```
from scipy.stats import norm
from math import log, sqrt, exp
def BS_EuroCall(S,K,r,q,sigma,T):
    d1=(log(S/K)+(r-q+sigma**2/2.)*T)/(sigma*sqrt(T))
    d2=d1-sigma*sqrt(T)
    c=S*exp(-q*T)*norm.cdf(d1)-K*exp(-r*T)*norm.cdf(d2)
    return c
```

012

BS_EuroCall.m

```
function c=BS_EuroCall(S,K,r,q,sigma,T)
d1=(log(S/K)+(r-q+sigma^2/2)*T)/(sigma*sqrt(T));
d2=d1-sigma*sqrt(T);
c=S*exp(-q*T)*normcdf(d1)-K*exp(-r*T)*normcdf(d2);
end
```

013

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 3: Given $S = 490$, $K = 470$, $r = 0.033$, $q = 0$, $T = 0.08$ and $c = 24.5941$, use function `fzero` and `BS_EuroCall` to solve the Black-Scholes equation for the implied volatility (σ).

013

Example 3: Given $S = 490$, $K = 470$, $r = 0.033$, $q = 0$, $T = 0.08$ and $c = 24.5941$, use `scipy.optimize.fsolve` / `myfzero` and `BS_EuroCall` to solve the Black-Scholes equation for the implied volatilty (σ).


```
[0.20000019]
```

012

```
0.20000018572007436
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

functools.partial

```
S=490
K=470
r=0.033
q=0
T=0.08
BS1=lambda x: BS_EuroCall(S, K, r, q, x, T)
BS1(0.2)
```

```
24.5940926130354
```

```
from functools import partial
S=490
K=470
r=0.033
q=0
T=0.08
c=24.5941
BS2=partial(BS_EuroCall, S, K, r, q, T=T)
BS2(0.2)
```

```
24.5940926130354
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

partial ⇨ mypartial?

`functools.`**`partial`**(*func[,*args][, **keywords]*)

Return a new partial object which when called will behave like *func* called with the positional arguments *args* and keyword arguments *keywords*. If more arguments are supplied to the call, they are appended to *args*. If additional keyword arguments are supplied, they extend and override *keywords*. Roughly equivalent to:

```
def partial(func, *args, **keywords):
    def newfunc(*fargs, **fkeywords):
        newkeywords = keywords.copy()
        newkeywords.update(fkeywords)
        return func(*(args + fargs), **newkeywords)
    newfunc.func = func
    newfunc.args = args
    newfunc.keywords = keywords
    return newfunc
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

```
sigma_imp=fsolve(lambda x: BS2(x)-c, 0.5)
print(sigma_imp)
```

```
[0.20000019]
```

014

```
def mypartial(fun, *args, **kwargs):
    def f(x):
        return fun(*args, x, **kwargs)
    return f
BS3=mypartial(BS_EuroCall, S, K, r, q, T=T)
c=BS3(0.2)
sigma_imp=fsolve(lambda x: BS3(x)-c, 0.5)
print(sigma_imp)
```

```
[0.2]
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 4: Compute Implied Volatility for Every Row in **`data`**

`dataset01.csv`

	A	B	C	D	E	F	G
1	S	K	r	q	sigma	T	c
2	1403	1350	5.34%	1.180%	0.26	0.10278	80.828
3	1403	1375	5.34%	1.180%	0.267	0.10278	66.084
4	1403	1400	5.34%	1.180%	0.231	0.10278	45.894
5	1403	1425	5.34%	1.180%	0.213	0.10278	30.955
6	1403	1450	5.34%	1.180%	0.198	0.10278	19.224

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 4: Find the implied volatility for each row of a table.

1. Use `readtable` to get the data from file `dataset01.csv`. Name the data imported as `data`.

```
data=readtable('dataset01.csv');
disp(data)
```

S	K	r	q	sigma	T	c
1403	1350	0.0534	0.0118	0.26	0.10278	80.828
1403	1375	0.0534	0.0118	0.267	0.10278	66.084
1403	1400	0.0534	0.0118	0.231	0.10278	45.894
1403	1425	0.0534	0.0118	0.213	0.10278	30.955
1403	1450	0.0534	0.0118	0.198	0.10278	19.224

2. Compute implied volatility for each row, and put the result in a new column with the column name `ImpVol`.

? How to apply a function to each row
? How to add a new column

3. Write `data` to a CSV file `output.csv`.

```
writetable(data,'output.csv');
```

015

```
data =
     S       K       r       q       sigma       T        c       ImpVol
    ____    ____    _____   _____    _____    _______   ______    _____

    1403    1350    0.0534  0.0118    0.26    0.10278   80.828     0.26
    1403    1375    0.0534  0.0118   0.267    0.10278   66.084    0.267
    1403    1400    0.0534  0.0118   0.231    0.10278   45.894    0.231
    1403    1425    0.0534  0.0118   0.213    0.10278   30.955    0.213
    1403    1450    0.0534  0.0118   0.198    0.10278   19.224    0.198
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 4: Compute the implied volatility for each row of `data`.

```
import pandas as pd
from scipy.optimize import fsolve
#Load data from dataset01.csv
data=pd.read_csv('dataset01.csv')
data
```

	S	K	r	q	sigma	T	c
0	1403	1350	0.0534	0.0118	0.260	0.102778	80.828
1	1403	1375	0.0534	0.0118	0.267	0.102778	66.084
2	1403	1400	0.0534	0.0118	0.231	0.102778	45.894
3	1403	1425	0.0534	0.0118	0.213	0.102778	30.955
4	1403	1450	0.0534	0.0118	0.198	0.102778	19.224

016


```
#Compute Implied Volatility for each row, and put the result in a new column
```

Write Your Answer Here.

```
data
```

	S	K	r	q	sigma	T	c	ImpVol
0	1403	1350	0.0534	0.0118	0.260	0.102778	80.828	0.260001
1	1403	1375	0.0534	0.0118	0.267	0.102778	66.084	0.266999
2	1403	1400	0.0534	0.0118	0.231	0.102778	45.894	0.230999
3	1403	1425	0.0534	0.0118	0.213	0.102778	30.955	0.212997
4	1403	1450	0.0534	0.0118	0.198	0.102778	19.224	0.198000

? How to “apply” a function to each row

? How to add a new column

```
data.to_csv('output.csv')
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Solve System of Nonlinear Equations

$$\begin{cases} f_1(x_1, \dots, x_m) = 0 \\ f_2(x_1, \dots, x_m) = 0 \\ \vdots \\ f_n(x_1, \dots, x_m) = 0 \end{cases}$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 5: Solve the following system of nonlinear equations

$$\begin{cases} x_0 \cos(x_1) = 4 \\ x_0 x_1 - x_1 = 5 \end{cases}$$

017

$$\Rightarrow \begin{cases} x_0 \cos(x_1) - 4 = 0 \\ x_0 x_1 - x_1 - 5 = 0 \end{cases}$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

```
def f03(x):
    y=[0, 0]
    y[0]=x[0]*cos(x[1])-4
    y[1]=x[0]*x[1]-x[1]-5
    return y
```

(Dr. Z: In Python, memory needs to be allocated before using.)

017

```
function y=f03(x)
    y(1)=x(1)*cos(x(2))-4;
    y(2)=x(1)*x(2)-x(2)-5;
end
```

(Dr. Z: In MATLAB, memory can be automatically allocated.)

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 5: Solve the following system of nonlinear equations for x_0 and x_1.

$$\begin{cases} x_0 \cos(x_1) = 4 \\ x_0 x_1 - x_1 = 5 \end{cases}$$

```
x=fsolve(@f03, [1,1])
```

```
x = 1x2 double
    6.5041    0.9084
```

017

fsolve

fsolve

Example 5: Solve the following system of nonlinear equations.

$$x_0 \cos(x_1) = 4$$
$$x_0 x_1 - x_1 = 5$$

```
from math import cos
def f03(x):
    y=[0, 0]
    y[0] = x[0]*cos(x[1]) - 4
    y[1] = x[0]*x[1] - x[1] - 5
    return y

x = fsolve(f03, [1, 1])
print(x)
```

```
[6.50409711 0.90841421]
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Optimization (see QF625, Session 03)

Minimize $f(\mathbf{x})$

w./w.o. constraints

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

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

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.

In general, the optimization problems are of the form:

```
minimize f(x) subject to

g_i(x) >= 0,  i = 1,...,m
h_j(x)  = 0,  j = 1,...,p
```

018

- ✓ fun
- ✓ x0
- ✓ bounds
- ✓ constraints

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

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

fmincon

R2018b

Find minimum of constrained nonlinear multivariable function

collapse all in page

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.

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(__)
```

019

- ✓ fun
- ✓ x0
- ✓ A
- ✓ b
- ✓ Aeq
- ✓ beq
- ✓ lb
- ✓ Ub
- ▪ nonlcon

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666 Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 6: Optimization with constraints

$$\min_x \; q(x) = (x_1 - 1)^2 + (x_2 - 2.5)^2$$

$$\begin{aligned}
\text{subject to} \quad & x_1 - 2x_2 + 2 \geq 0, \\
& -x_1 - 2x_2 + 6 \geq 0, \\
& -x_1 + 2x_2 + 2 \geq 0, \\
& x_1 \geq 0, \\
& x_2 \geq 0.
\end{aligned}$$

$x_1 \rightarrow$ x[0]
$x_2 \rightarrow$ x[1]

```
g_i(x) >= 0,
h_j(x)  = 0,
```

type : *str*

Constraint type: 'eq' for equality, 'ineq' for inequality.

fun : *callable*

The function defining the constraint.

```
from scipy.optimize import minimize
fun = lambda x: (x[0] - 1)**2 + (x[1] - 2.5)**2
cons = ({'type': 'ineq', 'fun': lambda x:  x[0] - 2 * x[1] + 2},
        {'type': 'ineq', 'fun': lambda x: -x[0] - 2 * x[1] + 6},
        {'type': 'ineq', 'fun': lambda x: -x[0] + 2 * x[1] + 2})
bnds = ((0, None), (0, None))
res = minimize(fun, (2, 0), bounds=bnds, constraints=cons)
print(res.x)
```

```
[1.4 1.7]
```

020

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 6: Optimization with constraints

$$\min_x \; q(x) = (x_1 - 1)^2 + (x_2 - 2.5)^2$$

$$\text{subject to} \quad x_1 - 2x_2 + 2 \geq 0,$$
$$-x_1 - 2x_2 + 6 \geq 0,$$
$$-x_1 + 2x_2 + 2 \geq 0,$$
$$x_1 \geq 0,$$
$$x_2 \geq 0.$$

020

$x_1 \rightarrow x(1)$
$x_2 \rightarrow x(2)$

$$c(x) \leq 0$$
$$ceq(x) = 0$$
$$A \cdot x \leq b$$
$$Aeq \cdot x = beq$$
$$lb \leq x \leq ub.$$

```
fun=@(x) (x(1)-1)^2+(x(2)-2.5)^2;
A=[-1 2;1 2;1 -2];
b=[2;6;2];
Aeq=[];
beq=[];
lb=[0 0];
ub=[];
x0=[0 0];
x= fmincon(fun,x0,A,b,Aeq,beq,lb,ub)
```

```
x =  1x2 double

    1.4000    1.7000
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

https://www.mathworks.com/help/matlab/ref/fminsearch.html

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

Unconstrained Optimization

R2018a

Solve unconstrained minimization problems in serial or parallel

Functions

`fminsearch`	Find minimum of unconstrained multivariable function using derivative-free method
`fminunc`	Find minimum of unconstrained multivariable function

Do Not Test

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

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

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

Constrained Optimization

R2018a

Solve constrained minimization and semi-infinite programming problems in serial or parallel

Solve problems using a modeling approach. Describe objective and constraints using symbolic variable expressions. For the steps to take, see Problem-Based Workflow.

Functions

fminbnd	Find minimum of single-variable function on fixed interval
fmincon	Find minimum of constrained nonlinear multivariable function
fseminf	Find minimum of semi-infinitely constrained multivariable nonlinear function

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

https://www.mathworks.com/help/optim/functionlist.html

Linear Programming and Mixed-Integer Linear Programming

Problem-Based Optimization

optimproblem	Create optimization problem
optimvar	Create optimization variables
showbounds	Display variable bounds
showproblem	Display optimization problem
showvar	Display optimization variable
writebounds	Save description of variable bounds
writeproblem	Save optimization problem description
writevar	Save optimization variable description
optimconstr	Create empty optimization constraint array
optimexpr	Create empty optimization expression array
showconstr	Display optimization constraint
showexpr	Display optimization expression
writeconstr	Save optimization constraint description
writeexpr	Save optimization expression description
evaluate	Evaluate optimization expression
findindex	Find numeric index equivalents of named index variables
infeasibility	Constraint violation at a point
prob2struct	Convert optimization problem to solver form
solve	Solve optimization problem
OptimizationConstraint	Optimization constraints
OptimizationExpression	Objective function or constraints
OptimizationProblem	Optimization problem
OptimizationVariable	Variable for optimization

Solver-Based Optimization

intlinprog	Mixed-integer linear programming (MILP)
linprog	Solve linear programming problems
mpsread	Read MPS file for LP and MILP optimization data

Linear Programming

Do Not Test

scipy.optimize.linprog

scipy.optimize.linprog(*c, A_ub=None, b_ub=None, A_eq=None, b_eq=None, bounds=None, method='simplex', callback=None, options=None*) [source]

Minimize a linear objective function subject to linear equality and inequality constraints.

Linear Programming is intended to solve the following problem form:

```
Minimize:     c^T * x

Subject to:   A_ub * x <= b_ub
              A_eq * x == b_eq
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

https://www.mathworks.com/help/optim/functionlist.html

Quadratic Programming

quadprog	Quadratic programming

Least Squares

Linear Least Squares

lsqlin	Solve constrained linear least-squares problems
lsqnonneg	Solve nonnegative linear least-squares problem
mldivide, \	Solve systems of linear equations Ax = B for x

Nonlinear Least Squares (Curve Fitting)

lsqcurvefit	Solve nonlinear curve-fitting (data-fitting) problems in least-squares sense
lsqnonlin	Solve nonlinear least-squares (nonlinear data-fitting) problems

Systems of Nonlinear Equations

fsolve	Solve system of nonlinear equations
fzero	Root of nonlinear function

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

quadprog

Quadratic programming

021

Solver for quadratic objective functions with linear constraints.

quadprog finds a minimum for a problem specified by

$$\min_x \frac{1}{2} x^T Hx + f^T x \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

H, A, and Aeq are matrices, and f, b, beq, lb, ub, and x are vectors.

You can pass f, lb, and ub as vectors or matrices; see Matrix Arguments.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

quadprog

Quadratic programming

Syntax

```
x = quadprog(H,f)
x = quadprog(H,f,A,b)
x = quadprog(H,f,A,b,Aeq,beq)
x = quadprog(H,f,A,b,Aeq,beq,lb,ub)
x = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0)
x = quadprog(H,f,A,b,Aeq,beq,lb,ub,x0,options)
x = quadprog(problem)
[x,fval] = quadprog(H,f,...)
[x,fval,exitflag] = quadprog(H,f,...)
[x,fval,exitflag,output] = quadprog(H,f,...)
[x,fval,exitflag,output,lambda] = quadprog(H,f,...)
```

Description

Finds a minimum for a problem specified by

$$\min_x \frac{1}{2}x^T Hx + f^T x \text{ such that } \begin{cases} A \cdot x \le b, \\ Aeq \cdot x = beq, \\ lb \le x \le ub. \end{cases}$$

$$\begin{pmatrix} x_1 & x_2 \end{pmatrix} \begin{pmatrix} a & c \\ c & b \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$= \begin{pmatrix} ax_1 + cx_2 & cx_1 + bx_2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

$$= (ax_1 + cx_2)x_1 + (cx_1 + bx_2)x_2$$

$$= ax_1^2 + bx_2^2 + 2cx_1x_2$$

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Examples

Solve a simple quadratic programming problem: find values of x that minimize

$$f(x) = \frac{1}{2}x_1^2 + x_2^2 - x_1x_2 - 2x_1 - 6x_2,$$

$$\frac{1}{2}(x_1^2 + 2x_2^2 - 2x_1x_2)$$

$$\frac{1}{2}(ax_1^2 + bx_2^2 + 2cx_1x_2)$$

$$H = \begin{bmatrix} a & c \\ c & b \end{bmatrix}$$

subject to

$$x_1 + x_2 \le 2$$
$$-x_1 + 2x_2 \le 2$$
$$2x_1 + x_2 \le 3$$
$$0 \le x_1,\ 0 \le x_2.$$

In matrix notation this is

$$f(x) = \frac{1}{2}x^T Hx + f^T x,$$

where

$$H = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix},\quad f = \begin{bmatrix} -2 \\ -6 \end{bmatrix},\quad x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}.$$

022

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 7: Quadratic Programming with linear constraints

$$f(x) = \frac{1}{2}x_1^2 + x_2^2 - x_1x_2 - 2x_1 - 6x_2$$

subject to the constraints

$$x_1 + x_2 \le 2$$
$$-x_1 + 2x_2 \le 2$$
$$2x_1 + x_2 \le 3.$$

022

```
H = [1 -1; -1 2];
f = [-2; -6];
A = [1 1; -1 2; 2 1];
b = [2; 2; 3];
x = quadprog(H,f,A,b)
```

```
x = 2x1 double

    0.6667
    1.3333
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 7: Optimization with constraints

$$f(x) = \frac{1}{2}x_1^2 + x_2^2 - x_1x_2 - 2x_1 - 6x_2$$

subject to the constraints

$$x_1 + x_2 \leq 2$$
$$-x_1 + 2x_2 \leq 2$$
$$2x_1 + x_2 \leq 3.$$

022

```
from scipy.optimize import minimize
fun = 
cons = 

bnds = ((None, None), (None, None))
res = minimize(fun, (0, 0), bounds=bnds, constraints=cons)
print(res.x)
```

```
[0.66666667 1.33333333]
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Minimum-Variance Portfolio

$$\min_{w} \quad w\Sigma w'$$
$$wR'=R_0$$
$$w_i \geq 0$$
$$\sum w_i = 1$$

4 assets

$$w = [w_1, w_2, w_3, w_4]$$
$$R = [0.07, 0.08, 0.09, 0.10], R_0 = 0.077$$
$$\Sigma = \begin{pmatrix} 0.0225 & 0.009 & 0.013125 & 0.01125 \\ 0.009 & 0.04 & 0.019 & 0.006 \\ 0.013125 & 0.019 & 0.0625 & 0.01125 \\ 0.01125 & 0.006 & 0.01125 & 0.09 \end{pmatrix}$$

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 8: Minimum-Variance Portfolio by `quadprog`

$$\min_{w} \quad w\Sigma w'$$
$$wR'=R_0$$
$$w_i \geq 0$$
$$\sum w_i=1$$

$w = [w_1, w_2, w_3, w_4]$

$R = [0.07, 0.08, 0.09, 0.10], R_0 = 0.077$

$$\Sigma = \begin{pmatrix} 0.0225 & 0.009 & 0.013125 & 0.01125 \\ 0.009 & 0.04 & 0.019 & 0.006 \\ 0.013125 & 0.019 & 0.0625 & 0.01125 \\ 0.01125 & 0.006 & 0.01125 & 0.09 \end{pmatrix}$$

```
V=
f=
A=
b=
Aeq=
beq=
lb=
ub=
w=quadprog(, f, A, b, Aeq, beq, lb, ub)
```

023

```
w = 4x1 double

    0.5705
    0.2615
    0.0655
    0.1025
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example 8: Minimum-Variance Portfolio

$$\min_{w} \quad w\Sigma w'$$

$$wR' = R_0$$

$$w_i \geq 0$$

$$\sum w_i = 1$$

023

$$w = [w_1, w_2, w_3, w_4]$$

$$R = [0.07, 0.08, 0.09, 0.10], R_0 = 0.077$$

$$\Sigma = \begin{pmatrix} 0.0225 & 0.009 & 0.013125 & 0.01125 \\ 0.009 & 0.04 & 0.019 & 0.006 \\ 0.013125 & 0.019 & 0.0625 & 0.01125 \\ 0.01125 & 0.006 & 0.01125 & 0.09 \end{pmatrix}$$

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

np.matrix

023

```
from scipy.optimize import minimize
import numpy as np
R=np.matrix([0.07,0.08,0.09,0.1])
x0=[0.1,0.1,0.1,0.1]
#x0=np.array([0.1,0.1,0.1,0.1])
#x0=np.array([[0.1,0.1,0.1,0.1]])
R0=0.077
sigma=np.matrix([[0.0225, 0.009, 0.013125, 0.01125],
                 [0.009, 0.04, 0.019, 0.006],
                 [0.013125, 0.019, 0.0625, 0.01125],
                 [0.01125, 0.006, 0.01125, 0.09]])
def fun(w, sigma):
    Mw=np.matrix(w)
    return (Mw*sigma*(Mw.T))[0,0]

cons = ({'type': 'eq', 'fun': lambda x: (np.matrix(x)*(R.T))[0,0]-R0},
        {'type': 'eq', 'fun': lambda x: np.sum(x)-1.0})
bnds = ((0, 1), )*4

res = minimize(fun, x0, args=sigma, bounds=bnds, constraints=cons)
res.x
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

np.array

023

```
from scipy.optimize import minimize
import numpy as np
R=np.array([[0.07,0.08,0.09,0.1]])
x0=[0.1,0.1,0.1,0.1]
#x0=np.array([0.1,0.1,0.1,0.1])
#x0=np.array([[0.1,0.1,0.1,0.1]])
R0=0.077
sigma=np.array([[0.0225, 0.009, 0.013125, 0.01125],
                [0.009, 0.04, 0.019, 0.006],
                [0.013125, 0.019, 0.0625, 0.01125],
                [0.01125, 0.006, 0.01125, 0.09]])
def fun(w, sigma):
    return (w@sigma@(w.T))

cons = ({'type': 'eq', 'fun': lambda x: (x@(R.T))[0]-R0},
        {'type': 'eq', 'fun': lambda x: np.sum(x)-1.0})
bnds = ((0, 1), )*4

res = minimize(fun, x0, args=sigma, bounds=bnds, constraints=cons)
res.x
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

np.array

023

```
from scipy.optimize import minimize
import numpy as np
R=np.array([0.07,0.08,0.09,0.1])
x0=[0.1,0.1,0.1,0.1]
#x0=np.array([0.1,0.1,0.1,0.1])
#x0=np.array([[0.1,0.1,0.1,0.1]])
R0=0.077
sigma=np.array([[0.0225, 0.009, 0.013125, 0.01125],
                [0.009, 0.04, 0.019, 0.006],
                [0.013125, 0.019, 0.0625, 0.01125],
                [0.01125, 0.006, 0.01125, 0.09]])
def fun(w, sigma):
    return (w@sigma@(w.T))

cons = ({'type': 'eq', 'fun': lambda x: (x@(R.T))-R0},
        {'type': 'eq', 'fun': lambda x: np.sum(x)-1.0})
bnds = ((0, 1), )*4

res = minimize(fun, x0, args=sigma, bounds=bnds, constraints=cons)
res.x
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

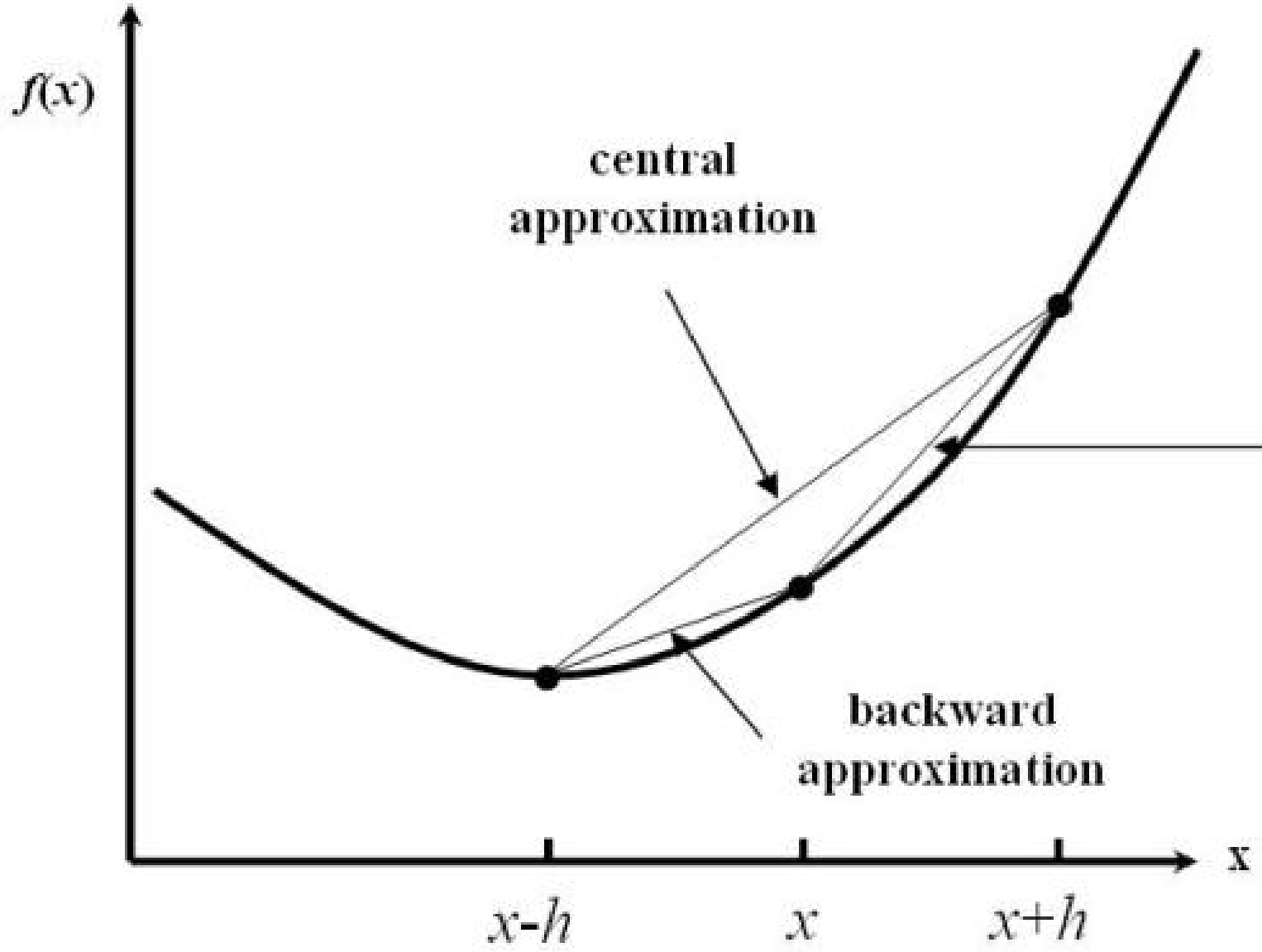
Differentiation

$$\frac{\partial}{\partial x_i} f(x_1, \ldots, x_n)\Big|_{\mathbf{x}=\mathbf{x}_0}$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance



$$f'(x) \approx \frac{f(x+h) - f(x)}{h}$$

$$f'(x) \approx \frac{f(x) - f(x-h)}{h}$$

$$f'(x) \approx \frac{f(x+h) - f(x-h)}{2h}$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

https://docs.scipy.org/doc/scipy/reference/generated/scipy.misc.derivative.html

scipy.misc.derivative

scipy.misc.derivative(*func, x0, dx=1.0, n=1, args=(), order=3*) [source]

Find the n-th derivative of a function at a point.

Given a function, use a central difference formula with spacing *dx* to compute the *n*-th derivative at *x0*.

Required arguments: **`func`**, **`x0`**, **`b`**

Default parameter value: **`dx=1.0`**

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

https://www.mathworks.com/help/matlab/ref/diff.html

diff

Differences and Approximate Derivatives

Syntax

```
Y = diff(X)
Y = diff(X,n)
Y = diff(X,n,dim)
```

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Approximate Derivatives with diff

Use the `diff` function to approximate partial derivatives with the syntax `Y = diff(f)/h`, where `f` is a vector of function values evaluated over some domain, `X`, and `h` is an appropriate step size.

Try This Example

For example, the first derivative of `sin(x)` with respect to `x` is `cos(x)`, and the second derivative with respect to `x` is `-sin(x)`. You can use `diff` to approximate these derivatives.

(Dr. Z: This approach is based on the finite difference method.)

```
h = 0.001;          % step size
X = -pi:h:pi;       % domain
f = sin(X);         % range
Y = diff(f)/h;      % first derivative
Z = diff(Y)/h;      % second derivative
plot(X(:,1:length(Y)),Y,'r',X,f,'b', X(:,1:length(Z)),Z,'k')
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

$$\Delta = \frac{\partial V}{\partial S} \qquad v = \frac{\partial V}{\partial \sigma}$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

https://en.wikipedia.org/wiki/Greeks_(finance)

$$\Delta = e^{-qT}\Phi(d_1), V = Se^{-qT}\phi(d_1)\sqrt{T}$$

```
import pandas as pd
from scipy.stats import norm
from scipy.misc import derivative
from math import log, sqrt, exp
def BS_EuroCallV(S,K,r,q,sigma,T):
    d1=(log(S/K)+(r-q+sigma**2/2)*T)/(sigma*sqrt(T))
    d2=d1-sigma*sqrt(T)
    c=S*exp(-q*T)*norm.cdf(d1)-K*exp(-r*T)*norm.cdf(d2)
    return c
def BS_EuroCallDelta(S,K,r,q,sigma,T):
    d1=(log(S/K)+(r-q+sigma**2/2)*T)/(sigma*sqrt(T))
    Delta=exp(-q*T)*norm.cdf(d1)
    return Delta
def BS_EuroCallVega(S,K,r,q,sigma,T):
    d1=(log(S/K)+(r-q+sigma**2/2)*T)/(sigma*sqrt(T))
    Vega=S*exp(-q*T)*norm.pdf(d1)*sqrt(T)
    return Vega
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

```
data=pd.read_csv('dataset01.csv',header=0)
data['BS']=data[['S','K','r','q','sigma','T']].apply(lambda x: BS_EuroCallV(*x), axis=1)
data['Delta']=data[['S','K','r','q','sigma','T']].apply(lambda x: BS_EuroCallDelta(*x), axis=1)
data['Delta2']=data[['S','K','r','q','sigma','T']].apply(
    lambda x: derivative(lambda s: BS_EuroCallV(s,*(x[1:])),x[0],dx=0.01), axis=1)
data['Vega']=data[['S','K','r','q','sigma','T']].apply(lambda x: BS_EuroCallVega(*x), axis=1)
data['Vega2']=data[['S','K','r','q','sigma','T']].apply(
    lambda x: derivative(lambda s: BS_EuroCallV(*(x[:4]),s,x[5]),x[4],dx=0.01), axis=1)
data
```

	S	K	r	q	sigma	T	c	BS	Delta	Delta2	Vega	Vega2
0	1403	1350	0.0534	0.0118	0.260	0.102778	80.828	80.827847	0.709677	0.709677	153.643372	153.615931
1	1403	1375	0.0534	0.0118	0.267	0.102778	66.084	66.084173	0.627880	0.627880	169.821334	169.811796
2	1403	1400	0.0534	0.0118	0.231	0.102778	45.894	45.894142	0.548545	0.548545	177.856491	177.855164
3	1403	1425	0.0534	0.0118	0.213	0.102778	30.955	30.955446	0.447307	0.447307	177.688245	177.682856
4	1403	1450	0.0534	0.0118	0.198	0.102778	19.224	19.224057	0.336832	0.336832	164.091086	164.051102

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Integration

$$\int_a^b f(x)dx$$

$$\int_a^b \int_{g(x)}^{h(x)} f(x,y)dy\,dx$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

scipy.integrate.quad

scipy.integrate.quad(*func, a, b, args=(), full_output=0, epsabs=1.49e-08, epsrel=1.49e-08, limit=50, points=None, weight=None, wvar=None, wopts=None, maxp1=50, limlst=50*) [source]

Compute a definite integral.

Integrate func from *a* to *b* (possibly infinite interval) using a technique from the Fortran library QUADPACK.

Required arguments: **func**, **a**, **b**

Returns:

y : *float*

The integral of func from *a* to *b*.

abserr : *float*

An estimate of the absolute error in the result.

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

integral

R2018b

Numerical integration

collapse all in page

Syntax

```
q = integral(fun,xmin,xmax)
q = integral(fun,xmin,xmax,Name,Value)
```

Description

example

`q = integral(fun,xmin,xmax)` numerically integrates function `fun` from `xmin` to `xmax` using global adaptive quadrature and default error tolerances.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example ?: Compute the following definite integral

$$\int_0^4 x^2 dx$$

```
from scipy import integrate
x2 = lambda x: x**2
print(integrate.quad(x2, 0, 4))
print(4**3 / 3.)  # analytical result
```

```
(21.333333333333336, 2.368475785867001e-13)
21.333333333333332
```

Use element-wise operators.

Example ?: Compute $\int_0^4 x^2 dx$.

```
x2=@(x) x.^2;
p=integral(x2, 0, 4)
```

```
p = 21.3333
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example ?: Compute the following definite integral

$$\int_0^{\infty} e^{-x} dx$$

```
from scipy import integrate
import numpy as np
invexp = lambda x: np.exp(-x)
integrate.quad(invexp, 0, np.inf)
```

```
(1.0000000000000002, 5.842606996763696e-11)
```

Example ?: Compute $\int_0^{\infty} e^{-x} dx$.

```
f=@(x) exp(-x);
p=integral(f, 0, inf)
```

```
p = 1
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

scipy.integrate.dblquad

scipy.integrate.dblquad(*func, a, b, gfun, hfun, args=(), epsabs=1.49e-08, epsrel=1.49e-08*) [source]

Compute a double integral.

Return the double (definite) integral of `func(y, x)` from `x = a..b` and `y = gfun(x)..hfun(x)`.

Required arguments: **func**, **a**, **b**, **gfun**, **hfun**

Returns:

y : *float*

The resultant integral.

abserr : *float*

An estimate of the error.

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

integral2

R2018b

Numerically evaluate double integral

collapse all in page

Syntax

```
q = integral2(fun,xmin,xmax,ymin,ymax)
q = integral2(fun,xmin,xmax,ymin,ymax,Name,Value)
```

Description

example

`q = integral2(fun,xmin,xmax,ymin,ymax)` approximates the integral of the function `z = fun(x,y)` over the planar region `xmin ≤ x ≤ xmax` and `ymin(x) ≤ y ≤ ymax(x)`.

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

Example ?: Compute the following definite integral

$$\int_0^2 \int_0^1 xy^2 dydx$$

SMU
SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance

```
from scipy import integrate
f = lambda y, x: x*y**2
integrate.dblquad(f, 0, 2, lambda x: 0, lambda x: 1)
```

```
(0.6666666666666667, 7.401486830834377e-15)
```

Even it is a constant in the inner integration's lower/upper limit, we need to use a constant function.

Example ?: Compute $\int_0^2 \int_0^1 xy^2 dydx.$

When it is a constant in the inner integration's lower/upper limit, we need to use the number. Constant function is not accepted.

```
f=@(x,y) x.*(y.^2);
p=integral2(f, 0, 2, 0, 1)
```

```
p = 0.6667
```

SMU SINGAPORE MANAGEMENT UNIVERSITY

QF666
Programming and Computational Finance

Dr. Zhao Yibao
Senior Lecturer
Of Quantitative Finance