FMO6 — Web: https://tinyurl.com/ycaloqk6 Polls: https://pollev.com/johnarmstron561

Numerical and Computational Methods in Finance

Dr John Armstrong

King's College London

August 22, 2020

Introduction

Course organization

- 2 hour lecture. This will define the course. (2 identical lectures per week)
- Exercises each week.
- 1 hour class. This is optional.
- 80% exam there are lots of past questions and papers on Keats.
- 20% coursework. This will be set in detail in **approximately** week 6.
- There is a quiz each week on Keats, plus exercise sheets on the course web page.

Why study FM06?

- Numbers are what matter.
- Required for most quant finance jobs.
- Required for the dissertation.
- Because numerical methods are actually fun.
    - Charts
    - Experiments
    - Insight
    - Bloomberg

What you will learn

- MATLAB. But why MATLAB?
  - Easy
  - Designed for science
  - Expected by employers.
- Monte Carlo methods
  - Simulate trading
  - Calculate risks
  - Price derivatives
- Other pricing methods
  - Solving the Black Scholes PDE
  - Pricing with trees
- Optimization
  - Portfolio optimization
  - Calibration

FM06 or FM13?

There is also a course FM13 on `C++` programming. Which should you choose?

- FM06 is a pre-requisite for FM13
- Study FM13 if you want to learn `C++` specifically.

Prerequisites

- Some knowledge of continuous time financial mathematics, i.e. one of: FM02,FM04 or 6CCM338a
- I will assume you are familiar with
  - The Black-Scholes model
  - Stochastic differential equations including Itô's Lemma
  - The Feynman-Kac theorem

Performing calculations in MATLAB

The MATLAB user interface

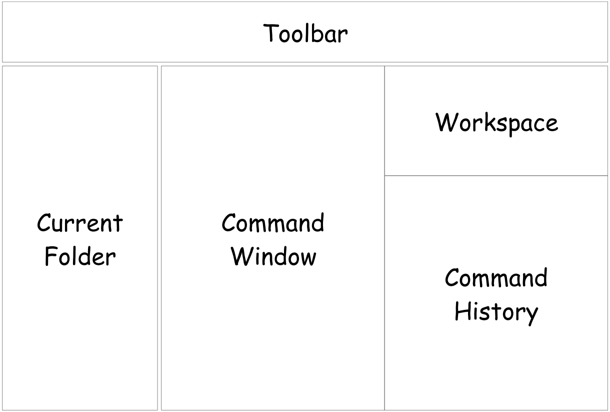

Figure: The layout of the MATLAB User Interface

Some simple commands

Enter the following in the **Command Window**

```
a = 3
b=2
a + b
```

- Checkout the **Workspace**.
- Checkout the **Command History**.

```
a = b + 25
a = a + 1
sin( 360 )
```

Using MATLAB for calculations

- Operators `*`, `+`, `-`, `^` and `/`.
- Functions just like in maths.
- Use brackets extensively.
- Use variables to keep your working.

Exercises

★ What is the cube root of 2?

★ Does `sin` use degrees or radian's?

★ What base is used for logarithms using the `log` command?

★ What happens if you forget the brackets and type `log 1`?

★ What happens if you type `a+1=a` instead of `a=a+1`?

★ Use the up arrow to run the command `a=a+1` repeatedly. Check that it is doing what you expect.

★ Work out how to compute 10 factorial.

Matrices

MATLAB = MATrix LABoratory.

$$A = \begin{pmatrix} 2 & 4 & 5 \\ -3 & 1 & 7 \\ 4 & 9 & 2 \end{pmatrix}$$

```
A = [ 2 4 5 ; -3 1 7 ; 4 9 2 ]
v1 = [ 1 2 3 ]
v2 = [ 4 5 6 ]
w1 = [1; 2; 3 ]
w2 = [4; 5; 6 ]
A * w1
w1 + w2
```

Punctuation

<table>
<tr><th>Symbol</th><th>Term</th></tr>
<tr><td>.</td><td>Full stop or just “dot”.</td></tr>
<tr><td>,</td><td>Comma.</td></tr>
<tr><td>:</td><td>Colon.</td></tr>
<tr><td>;</td><td>Semi-colon.</td></tr>
<tr><td>’</td><td>Apostrophe or single quote.</td></tr>
<tr><td>"</td><td>Double quote</td></tr>
<tr><td>_</td><td>Underscore.</td></tr>
<tr><td>()</td><td>Brackets, also called round brackets or parentheses.</td></tr>
<tr><td>[]</td><td>Square brackets.</td></tr>
<tr><td>{}</td><td>Curly brackets.</td></tr>
<tr><td><></td><td>Angle brackets.</td></tr>
<tr><td>~</td><td>Tilde or twiddle.</td></tr>
<tr><td>&</td><td>Ampersand or and sign.</td></tr>
<tr><td>|</td><td>Pipe or vertical line</td></tr>
</table>

Dividing matrices

To solve

$$Aw' = w$$

think

$$w' = A^{-1}w$$

so "divide on the left".
To solve

$$v'A = v$$

think

$$v' = vA^{-1}$$

so "divide on the right".

```
A \ w1
v1 / A
```

Inverse

```
inv(A)
```

Transpose

Use ' for the conjugate transpose.

```
w1 = [1 2 3]';
w2 = [4 5 6]';
```

Creating matrices

```
zeros(4,6)
rand(3,5)
randn(3,5)
zeros(4)
diag([2 4 7])
eye(5)
1:100
20:50
2:3:50
linspace(30,70,10)
```

Dot operators

```
dollarPrices = [ 100 105 103 102 103 ]
gbpToUsdRates = [ 0.61 0.62 0.63 0.62 0.61]
gbpPrices = dollarPrices .* gbpToUsdRates
```

- `.*` means elementwise multiplication
- `*` means matrix multiplication

Tip: Use long variable names

Note that you type faster than you think.

Statistical functions

```
sum(dollarPrices)
mean(dollarPrices)
length(dollarPrices)
std(dollarPrices) % Sample s.d.
sum(A)
prctile( dollarPrices, 25 )
```

Histograms

```
sample = randn(10000, 1)
hist( sample )
```

```
sample = randn(10000, 1);
hist( sample, 100 )
```

Tip: Semi-colons

A semi-colon at the end of a line means supress output.

Example

Use MATLAB to compute the sum

$$1 + 2 + 2^2 + 2^3 + 2^4 + \ldots + 2^{10}$$

```
x = 0:10;
powers = 2.^x;
sum( powers )
```

You can do it in one line `sum( 2.^(0:10) )`.

Example

A robot walks a distance X_1 east, a distance X_2 south and then climbs a distance X_3 up. The X_i are independent and normally distributed with mean 0 and standard deviation 1. Negative distances should be interpreted in the obvious way. Using a MATLAB simulation, plot an approximate histogram of the total distance travelled.

```
X1 = randn(1000,1);
X2 = randn(1000,1);
X3 = randn(1000,1);
distance = sqrt( X1.^2 + X2.^2 + X3.^2 );
hist( distance, 20 );
```

Exercises

Use MATLAB to answer the following questions:

★ What is $\left(\frac{1}{\sqrt{2}}(1+i)\right)^4$?

★ How would you create a vector containing the first 50 odd integers in MATLAB? What is the sum of the first 50 odd integers?

★ What is the 95-th percentile of the normal distribution (with mean 0 and standard deviation 1).

★ Recall that $\pi = 4(1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \ldots)$. Compute π to three decimal places. (I don't expect you to answer all of these in the time I'll give you.)

Summary

We can

- Use MATLAB as a sophisticated calculator.
- Use *variables* to store our data.
- Use `*`, `+`, `^`, `/` etc. with numbers and matrices.
- Understand the difference between `*` and `.*`
- Create matrices with the `zeros`, `eye`, `randn` etc..
- Compute statistics with `std`, `mean`, `length`, `hist`

Functions

Some functions we would like to write

(i) `cumulativeNormal(x)`. Computes $\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-t^2/2)\,\mathrm{d}t$.

(ii) `blackScholesCallPrice( K, T, S, vol, r)`.

(iii) `integrateNumerically( f, a, b, N)`. Returns an approximation to $\int_a^b f(t)\,\mathrm{d}t$ computed using the rectangle method with N steps.

Key terms: parameter, return value.

MATLAB functions are a little different to maths functions because not only can they return a value, they can do something. E.g. `hist`.

- Functions allow us to solve a problem once and then reuse the solution. Here is a *deliberately difficult* problem we wouldn't want to solve repeatedly.
- How can we compute $N(x)$ the cumulative distribution function of the normal distribution? One answer is to use the built in function `normcdf`, but suppose that didn't exist? How could we write our own function?
- IDEA: Make the substitution $t = x + 1 - \frac{1}{s}$ to transform the integral
$$\int_{-\infty}^{x} \exp(-t^2/2)\,\mathrm{d}t$$
to an integral of a finite interval. Then use the rectangle rule.
- Writing the code to do this will be tricky, but functions will allow us to write our code so it can solve the problem for any value of x without us needing to think.

Substitution

Write

$$t = x + 1 - \frac{1}{s}$$

So

$$\frac{\mathrm{d}t}{\mathrm{d}s} = \frac{1}{s^2}$$

and as $s \to 1$, $t \to x$ whereas as $s \to 0$, $t \to -\infty$. Putting this together we find:

$$\frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-t^2/2)\,\mathrm{d}t = \frac{1}{\sqrt{2\pi}} \int_0^1 \frac{1}{s^2} \exp\left(-\frac{1}{2}\left(x + 1 - \frac{1}{s}\right)^2\right) \mathrm{d}s$$

The Rectangle Rule

$f : [a, b] \longrightarrow \mathbb{R}$. Approximate f with N rectangles to compute integral. Define

$$\begin{aligned} h &= \frac{b-a}{N} \\ s_n &= a + (n - \frac{1}{2})h \end{aligned}$$

then

$$\int_a^b f(s)\,\mathrm{d}s \approx h \sum_1^N f(s_n).$$

To solve the problem take

$$f(s) = \frac{1}{s^2} \exp(-(x + 1 - \frac{1}{s})^2/2)$$

and $a = 0$, $b = 1$ and $N = 1000$ (say)

Complete mathematical solution

$$
\begin{aligned}
N(x) &= \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} \exp(-t^2/2)\,\mathrm{d}t \\
&= \frac{1}{\sqrt{2\pi}} \int_{0}^{1} \frac{1}{s^2} \exp\left(-\frac{1}{2}\left(x+1-\frac{1}{s}\right)^2\right)\,\mathrm{d}s \\
&\approx \frac{h}{\sqrt{2\pi}} \sum_{n=1}^{N} \frac{1}{s_n^2} \exp\left(-\frac{1}{2}\left(x+1-\frac{1}{s_n}\right)^2\right)
\end{aligned}
$$

where

$$
a=0, \quad b=1, \quad N=1000, h=\frac{b-a}{N}
$$

and

$$
s_n = a + (n-\frac{1}{2})h
$$

MATLAB solution

```
x = 1.5;
a = 0;
b = 1;
N = 1000;
h = (b-a)/N;
s =  a + ((1:N) - 0.5) * h;
fValues = s.^(-2) .* ...
    exp( -(( x + 1 - 1./s).^2)/2 );
integral = h * sum( fValues );
result = 1 / sqrt( 2 * pi ) * integral
```

Tip: `Use␣...␣for␣long␣lines`

Why functions?

Problem:

- We had to use a lot of brain power
- The code is hard to follow
- We don't want to have to type all this every time we need to calculate the cumulative normal distribution.
- We want to save our work to a file

The solution? Functions.

Creating a function

(i) Use Windows Explorer to create a folder `FM06` in your home area.
(ii) Create a sub folder called `Lecture1`
(iii) In MATLAB set your current folder to `FM06/Lecture1`
(iv) Right click in the **Current Folder** and select **New File→Function**
(v) Type the name of the function. This should be `cumulativeNormal.m` **PRECISELY**.
(vi) You've now created a file.
(vii) Edit the file.

An example function

In the **editor window** replace all text with:

```
function [ result ] = cumulativeNormal( x )
%CUMULATIVENORMAL computes c.d.f of normal distribution
a = 0;
b = 1;
N = 1000;
h = (b-a)/N;
s =  a + ((1:N) - 0.5) * h;
fValues = s.^(-2) .* exp( -(( x + 1 - 1./s).^2)/2 );
integral = h * sum( fValues );
result = 1 / sqrt( 2 * pi ) * integral;
end
```

Save it. Run it.

Save the file. Run the function with:

```
cumulativeNormal( 1.5 );
```

In general

The syntax is:

```
function [ <output values> ] = ...
  <function Name>( <input values> )
```

Tip: Check list

- Is the function name exactly the same as the file?
- Have you saved the file?
- Have you selected the correct current folder.
- Have you got rid of all red marks?

Another example

Example

Write a function to convert from polar coordinates to Cartesian coordinates.

```
function [ x, y ] = polarToCartesian( r, theta )
x = r * cos( theta );
y = r * sin( theta );
end
```

Using a function with multiple return values

```
r = 2.0;
theta = pi/2;
[ x, y ] = polarToCartesian( r, theta );
disp( x ); % Prints out the value of x
disp( y ); % Prints out the value of y

%If you don't need y
x = polarToCartesian( r, theta );

%If you don't need x
[~,y] = polarToCartesian( r, theta );
```

Homework

Complete the exercises on worksheet 1.