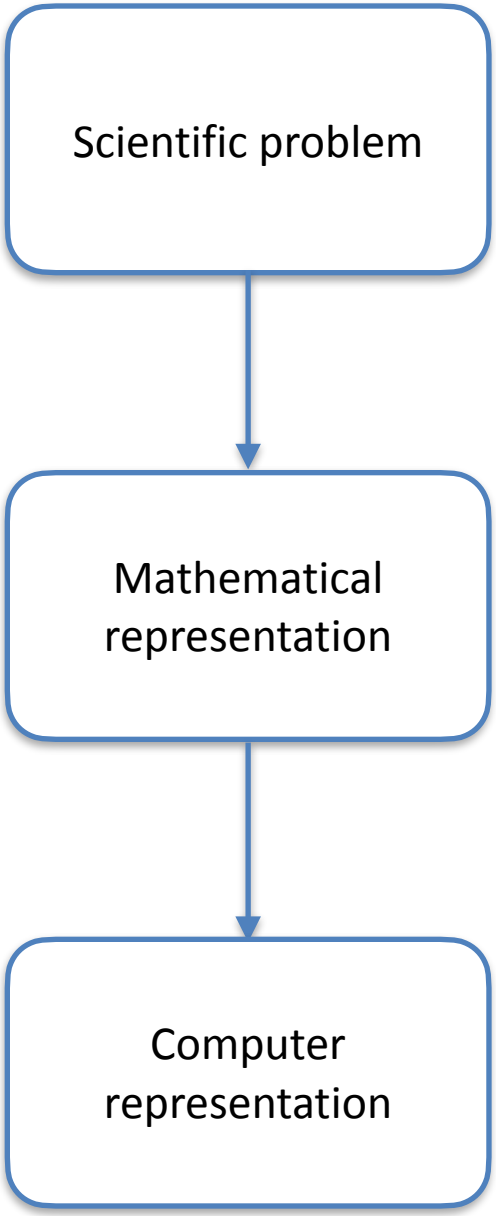
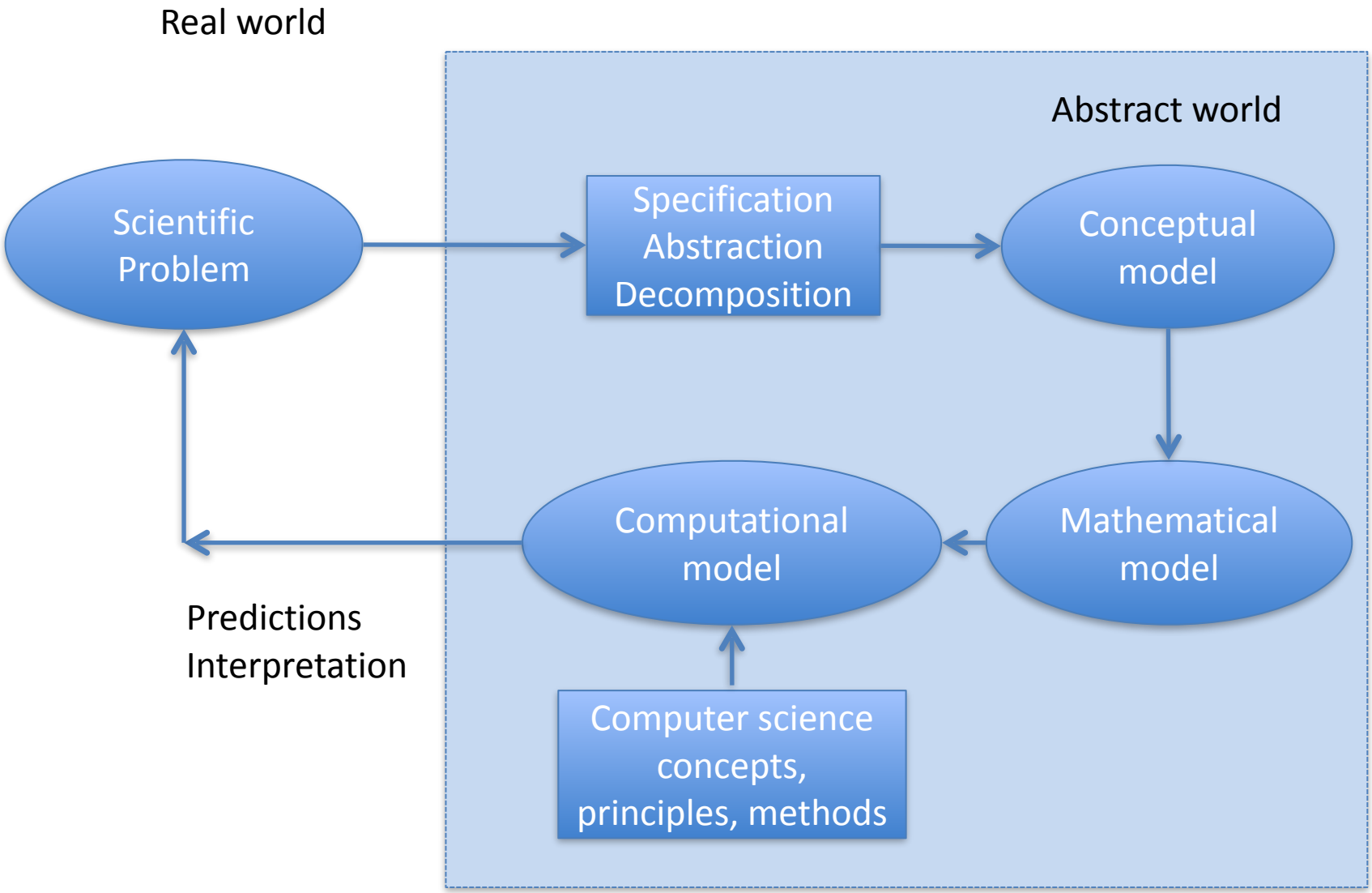
1. Introduction to Computer Modelling

1.1 Computer Models



Different levels of abstraction

Levels of abstraction



Computer problem solving workflow

i) Problem definition
ii) Model specification (parameters, constraints)
iii) Mathematical Model
iv) Implementation
v) Verification
vi) Validation

Task 1:

Install the latest python version on your computer/laptop using anaconda (this will include the most important libraries):

https://docs.anaconda.com/anaconda/install/windows/

(select python 3.7 – install – go to anaconda in start menu – click on jupyter notebook – select jupyter notebook)

Work out the example: "Problem 1" (download the file and open in jupyter notebook)

- use the circumference and area of a circle as an example
- prepare your own notebook to carry out the calculation
- provide detail about the above-mentioned steps i) to vi) as comments in the notebook
- upload the notebook to moodle

Different types of problems and models

continuous ("analytical")

discrete ("numerical")

the classical motion of an object - i.e. a car driving down the road; a swinging pendulum

"quantum" motion of a particle through "jumps" - i.e. Brownian motion of a particle on an ordered surface

the temperature of an object - i.e. a pizza in the oven; the heat generated through a chemical reaction

discrete states - i.e. protein folding

$f(x) = x^2$

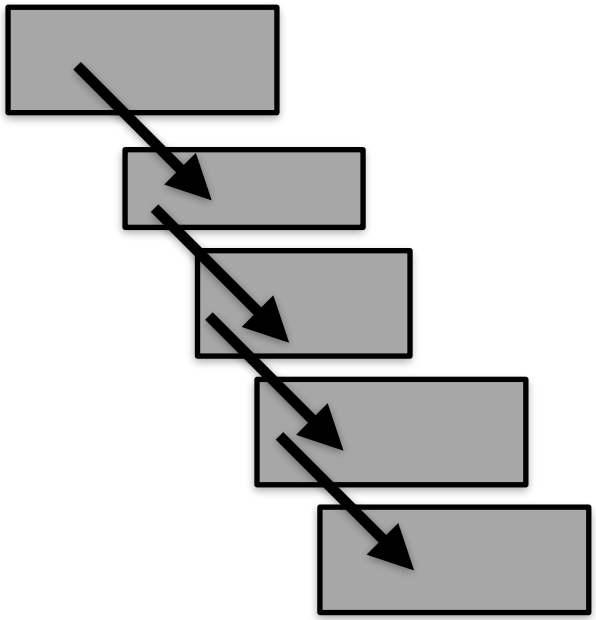
Integral?

$$\int f(x)dx = \frac{x^3}{3}$$

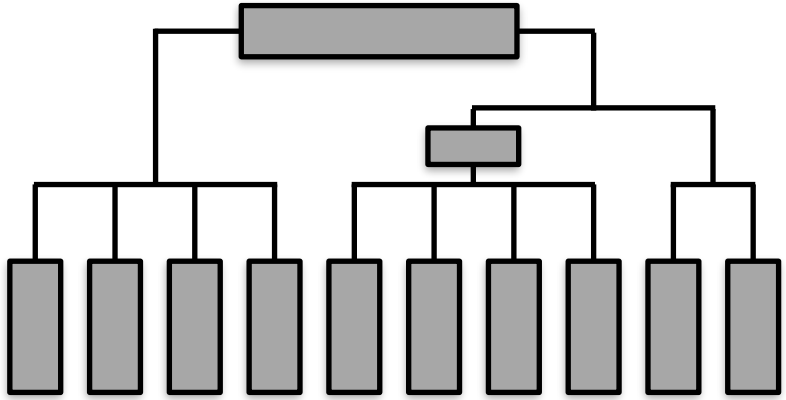
$$\int_b^b f(x)dx = \lim_{n\to\infty} \frac{b-a}{n} \sum_{k=1}^{n} f(a + k\frac{b-a}{n})$$

Problem decomposition

top-down
(i.e. “waterfall”)



modular
(i.e. “divide and conquer”)



Programming languages

High-level programming languages:
- Fortran, C, C++, Java, python
- hardware-independent, readable
- source program or source code

(i) compiler compiles the program into machine code (Fortran, C/C++)
(ii) interpreter reads and executes program (python)
(iii) compiler compiles the program into bytecode, which is then interpreted (Java)

- often, compiled program also needs to be linked to libraries
- compiled program is platform-dependent

Low-level programming languages:
- machine language, assembly language
- hardware-dependent, low readability

- program directly runs on processor

Python programming language

- founded in 1989 by Guido van Rossum; being a big fan of Monty Python's flying circus, he picked the name "python"
- Goals of python: easy and intuitive language, open source, understandable code, suitable for daily tasks
- comprehensive standard library

Syntax and properties:

- use of whitespace to denote blocks:

```
for i in range(10):
    x=m*i
```

- collection types - lists can contain any kind of objects
- use of dictionaries - fast lookup times, dictionaries are intrinsic to python objects

TIOBE Index for March 2020

March Headline: Delphi is about to fall out of the TIOBE index top 20

The end seems to be near for the well-beloved Delphi language. Delphi has been in the top 20 since the beginning of the TIOBE index (started in June 2001). In the early 2000s it was one of the most popular languages and IDEs. Borland Delphi 7 of 2002 was used by more Delphi developers than any other single version of Delphi. It was a complete environment to create Windows applications. After that Delphi got in to troubles: the port to Linux was not successful, there were some buggy releases and non-commercial IDEs with similar features started to conquer the market. In some countries, such as Brazil, Delphi remained very popular for a long time.

The TIOBE Programming Community index is an indicator of the popularity of programming languages. The index is updated once a month. The ratings are based on the number of skilled engineers world-wide, courses and third party vendors. Popular search engines such as Google, Bing, Yahoo!, Wikipedia, Amazon, YouTube and Baidu are used to calculate the ratings. It is important to note that the TIOBE index is not about the *best* programming language or the language in which *most lines of code* have been written.

The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system. The definition of the TIOBE index can be found here.

Mar 2020	Mar 2019	Change	Programming Language	Ratings	Change
1	1		Java	17.78%	+2.90%
2	2		C	16.33%	+3.03%
3	3		Python	10.11%	+1.85%
4	4		C++	6.79%	-1.34%
5	6	^	C#	5.32%	+2.05%
6	5	v	Visual Basic .NET	5.26%	-1.17%

Number representation

floating-point arithmetic: approximation of real number through finite digits

π = 3.14159265358979323846264338327950288419716939937510582097 4…
in double precision (occupying 64 bits): correct up to 16 digits
π = 3.14159265358979311599796346854418516159057617187 5…
in single precision (occupying 32 bits): correct up to 8 digits
π = 3.14159274101257324218 75…

overflow: occurs when value is too large to be represented with available digits; for example, for integers

2^{64} -1 = 18,446,744,073,709,551,615

for floats in python

```
print(sys.float_info.max)
```

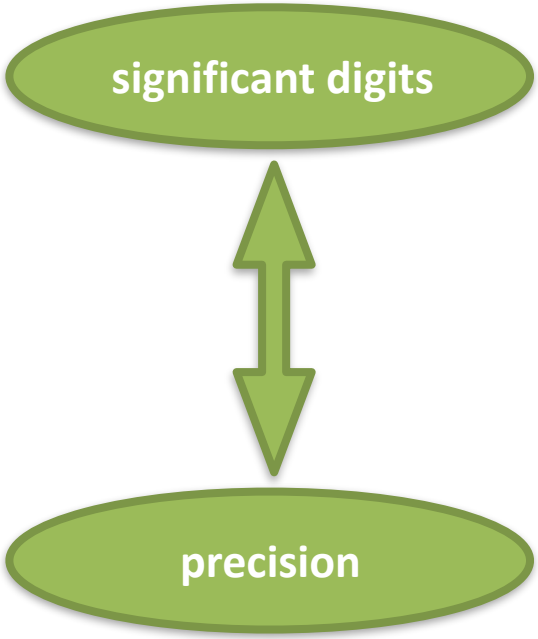
```
1.7976931348623157e+308
```

underflow: value is too small; for floats in python

```
: print(sys.float_info.min)
```

```
2.2250738585072014e-308
```

Some remarks



vs.

accuracy: from numerical approximation *

*in the following, we will focus on **numerical approaches**
but: precision is important

types of variables:

numbers (integers, floats, complex)	lists (tuples)
strings	dictionaries

Task 2:

- work out the example: “Problem 2”
- upload the notebook to moodle