

Exercise I - Mathematica Tutorial

General Information

The cycle of exercises will be:

- distribution of the new exercise sheet every tuesday via www.nano-optics.ethz.ch/education
- handing-in solved exercises every tuesday at HCI, F221, box: Signal Processing/Messtechnik
- discussion of solved and new exercises during the exercise classes

The assistants will be:

Martin Huppert	HCI F 214	huppertm@phys.chem.ethz.ch
Michael Batel	HCI D 222.1	miba@nmr.phys.chem.ethz.ch
Dr. Stephan Götzinger	HCI F 220	stephan.goetzing@phys.chem.ethz.ch

Working with *Mathematica*

Mathematica is a computational software package for both analytic as well as numerical calculations and graphical visualization. As a student at ETH, you can purchase Mathematica from IDES free of charge (ides.ethz.ch).

Mathematica has incorporated an extensive *Help* option. In *Help* → *Documentation Center* arbitrary functions can be searched and explored. Furthermore in *Help* → *Virtual Book* an educative introduction to *Mathematica* can be found. The following introduction should give you a first inside to the work with *Mathematica*.

Mathematica documents are called “notebooks” and consist of a sequence of pairs of input and output cells. A cell is where you enter a Mathematica command or receive output from the program. The notebook represents only one part of the program, its front end. Here the communication to the user occurs, it is the interface you will be working with. The actual calculations are done “under the hood” by another module called the kernel.

First steps

Open the *Mathematica* program on your computer. You should see an empty notebook without any cells. Clicking the mouse somewhere on the notebook will activate the cursor. A cell can run over multiple lines, in the case of an input cell it is terminated by hitting *Shift + Return* or simply the *Return* key on the number block of your keyboard. For writing comments that should not be interpreted by *Mathematica*, you can enclose your comment as: (** here your comment **).

Typing $(2+4)*5$ and hitting *Shift + Return* gives

```
In[1]:= (2 + 4) * 5
```

```
Out[1]= 30
```

The program takes the first line as input and returns the result as output, both as separate cells. In *Mathematica* we employ for adding, subtracting, multiplication, dividing, and exponentiating the symbols $+$, $-$, $*$, $/$, \wedge , respectively. Complex numbers are entered as $(x + I y)$. Instead of using the symbol $*$ for multiplication one also can just use a space between the factors.

As output result *Mathematica* chooses always the most accurate result. Hence typing

```
In[2]:= 12/28
```

```
Out[2]=  $\frac{3}{7}$ 
```

only simplifies the fraction. If a decimal result with accuracy of 3 digits is desired, use:

```
In[3]:= N[12/28, 3]
```

```
Out[3]= 0.429
```

The `N[]` command tells *Mathematica* to evaluate the quantity in brackets numerically. *Mathematica* comes with many build in commands. On a command call, variables and (optional) parameters are passed with square brackets. Note: *Mathematica* discriminates between capital and small letter! Build in *Mathematica* functions or commands usually start with capital letters.

Symbolic algebra

Mathematica can handle symbolic expressions. The expression

```
In[4]:= 1/(5 + (x - 2)^2)
```

```
Out[4]=  $\frac{1}{5+(-2+x)^2}$ 
```

can be differentiated by typing

```
In[5]:= D[%, x]
```

```
Out[5]=  $-\frac{2(-2+x)}{(5+(-2+x)^2)^2}$ 
```

where `D[]` is the method for taking the derivative (parameter `x` tells the method to which variable to differentiate). The `%` symbol refers to the last generated output.

You can assign expressions to variables:

```
In[6]:= f = E^(I * Pi * x)
```

```
Out[6]=  $e^{i\pi x}$ 
```

or simple values:

```
In[7]:= x = 1/6
```

```
Out[7]=  $\frac{1}{6}$ 
```

```
In[8]:= N[E^(I * Pi * x)]
```

```
Out[8]= 0.866025 + 0.5i
```

Here, first `x` is assigned `1/6`, then this value is fed to the above expression. In this example you can see that some of the strings have already a predefined meaning, like `E`, `Pi`, and `I`,

If one wants to evaluate a given expression at a certain argument, one can define custom functions. The *Mathematica* syntax for defining a function can be seen in the following example:

```
In[9]:= g[t.]:=E^(I * Pi * t)
```

the function is assigned to the symbol **g**, the argument passed when calling it is assigned to **t**. The underscore **_** is important and is used only on the left-hand side. Now the function can be evaluated at a given point:

```
In[10]:= g[1/2]
```

```
Out[10]= i
```

or for example be integrated from 0 to 1/2:

```
In[11]:= Integrate[g[t], {t, 0, 1/2}]
```

```
Out[11]=  $\frac{1+i}{\pi}$ 
```

Working with lists

Vectors and matrices are represented by *Mathematica* as lists. Elements of a list stand between curly brackets, separated by commas. The order of the elements determines the index by means of which one can refer to the elements:

```
In[12]:= mylist = {a, b, c}
```

```
Out[12]= {a, b, c}
```

```
In[13]:= mylist[[2]]
```

```
Out[13]= b
```

Matrices can be defined as nested lists:

```
In[14]:= mat = {{1, 2, 3}, {A, B, C}, {5, 6, 7}}
```

```
Out[14]= {{1, 2, 3}, {A, B, C}, {5, 6, 7}}
```

and be scalar multiplied with any list of correct dimensions (by means of known rules for scalar multiplication) using the dot operator **.** (period):

```
In[15]:= mat.mylist
```

```
Out[15]= {a + 2b + 3c, aA + bB + cC, 5a + 6b + 7c}
```

For numerical calculations it is often necessary to translate analytical expressions to discrete lists. In *Mathematica* this can be achieved with the function **Table[]**, for example for a cosine function with:

```
In[16]:= coslist = Table[Cos[x], {x, 0, 2Pi, Pi/10}];
```

Here, a list **coslist** is generated with 20 discrete and equal spaced cosine values between argument 0 and 2π . Note that the semicolon at the end of the line suppresses the output to be generated.

Problem 1:

Please conduct the following operations using *Mathematica* and its documentation to find the necessary functions and methods. Notebooks can be saved as pdf or printed directly for handing in your results.

1. Define a function:

$$f(x) = e^{-x}x^2$$

and evaluate at a given point to an exact as well as decimal expression with 5 digit precision.

2. Plot the function from 0 to 6π .
3. Give the indefinite integral of $f(x)$.
4. Integrate $f(x)$ numerically from $-\pi$ to π . There are two similar possibilities in *Mathematica* for this operation, what could be the difference?

Problem 2: Curve fitting

Define a list with the entries:

$$\{2.07, 1.36, 1.12, 1.29, 1.18, 1.37, 1.84, 1.90, 2.08, 2.17, 2.33, 2.58, 3.06, 3.08, 3.19\}$$

1. Plot the list using the `ListPlot[]` command in the range of $x=(0,15)$ and $y=(0,5)$
2. Find a fit for the data using `Fit[]` and assuming a linear combination of an exponential and a linear function.
3. Plot both, the fit and the list, in one window, again with same limits.

Problem 3: Solving analytical equations

The most powerful feature of *Mathematica* is its possibility to run analytical calculations. A remarkable method is the `Solve[]` method. Using this, solve the equation:

$$\sqrt{ax^2 + x - 1} * (2 - x)^2 = 0$$