



Mathematica - a First Glance

Peter Fischer, ZITI



What is Mathematica ?

- Initially: **algebraic** manipulation of formulae

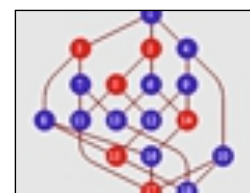
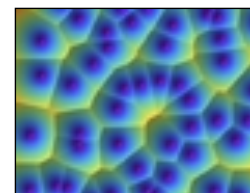
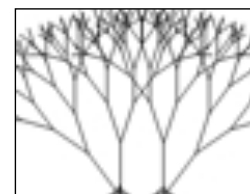
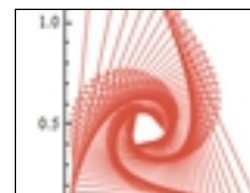
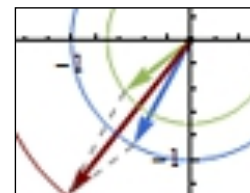
- Integration
- Differentiation
- Many special functions

- Later:

- Numerical methods
- Fancy graphics
- Sound, Image processing, lots of specialties...

- Recently:

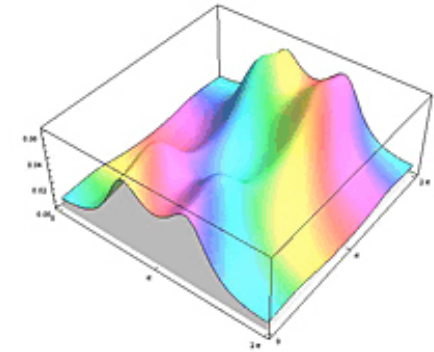
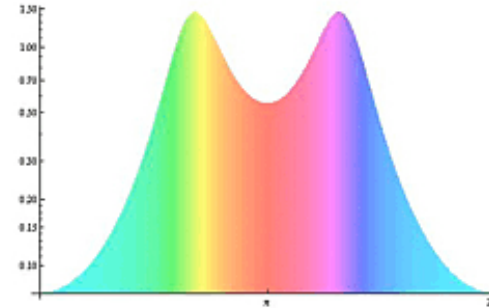
- 'Demonstrations' on the Web, see →
(notebooks can be downloaded)
- <http://demonstrations.wolfram.com/>
- Separate 'CDF player' to locally run demos





What can WE use it for

- Function Plotting
 - For publications
 - For our understanding
- Equation solving, optimizations
- Verification of manual results
- Simple Data display and analysis
- Educational display of relations / effects / ...
- For fun!





HOW can we use it ?

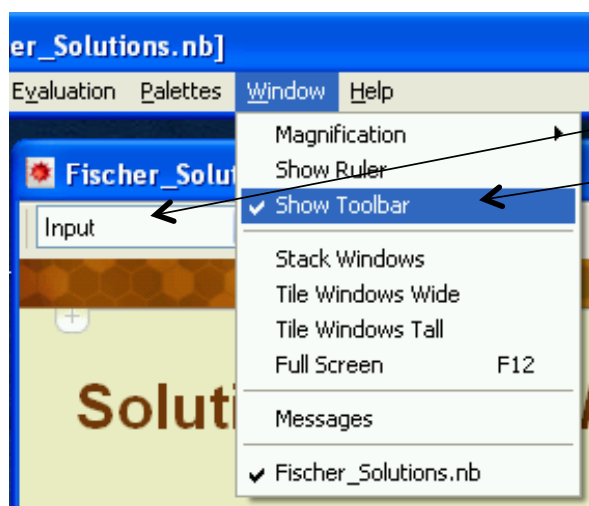
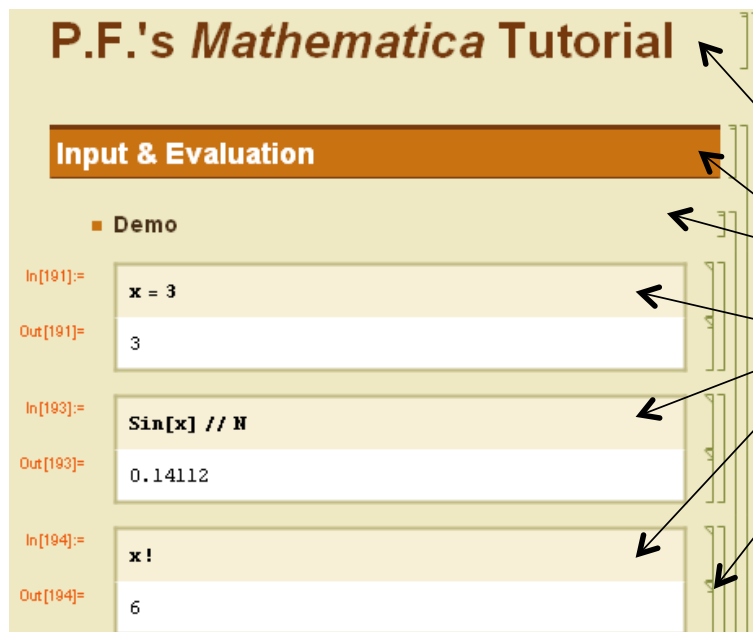
- Licenses are (very) expensive
- But: Mathematica is available on the CIP Pool Machines
- 15 day trial version is *free*!
- On the CIP Pool type:
 - > **mathematica** &
 - Sorry: only 20 (?) licenses available. Please use in teams!!!!
- Other similar tools:
 - **Maple**: commercial, www.maplesoft.com
 - **Magma**: commercial, magma.maths.usyd.edu.au/magma
Mainly Computer Algebra, no graphics.
 - **MATLAB**:...
more for matrices, i.e. linear systems
 - **Octave**: open source alternative to MATLAB
 - **sage**: open source (pronounce 'seetch') www.sagemath.org
(I have not tried it)



BASICS



Notebooks



- User interface: *Notebooks*
- Each group has a specific format
 - Document title, subtitle,...
 - Sections, subsection,...
 - Input - **This must be used for all input !**
- Hierarchical Input: Blocks can be made invisible by clicking on rulers on right side
- Colors / Styles use templates
 - *Format* → *StyleSheet* → ...
- Get format selection box in Window from
 - *Window* → *Show Toolbar*
- Before saving a notebook, the (memory consuming) outputs can be cleared with
 - *Cell* → *Delete All Output*
- File extension: *.nb



Entering Stuff

- An input is 'executed' by ending with **SHIFT - Return**
 - Just entering CR starts a new line but does *not* execute it
- The *input* line is evaluated and an *output* is produced
 - To disable output (for definitions...), end the input line with ';'
 - Each input/output line gets a line number
 - The last input line can be referred to as '%'
 - The line before with '%%' etc.
 - Any line can be referred to by its number with %n (no blank)

In[126]:=	Log[e]
Out[126]=	1
In[127]:=	x = 5;
In[128]:=	% + 2
Out[128]=	7

▪ Parenthesis

- (...) for normal mathematical priorities
- [...] for *function* arguments
- {...} are *lists* (see later), used also for vectors, matrices,...
- [[n]] are *indices* in lists (see later)
- (* ... *) are *comments*



Getting Help

- Select a key word, then press F1
- In the Help Window
 - Use key words
 - Use 'see also' section at the bottom
 - Very nice feature:
all examples can be executed and modified
- More help on wolfram pages, for instance
 - <http://www.wolfram.com/support/learn/>



Formatting Input

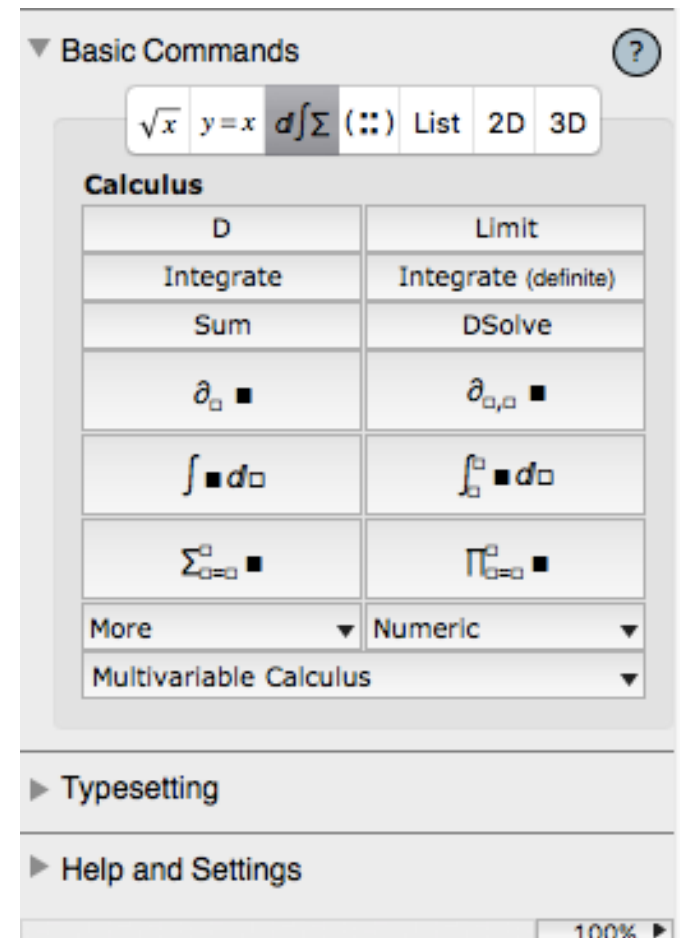
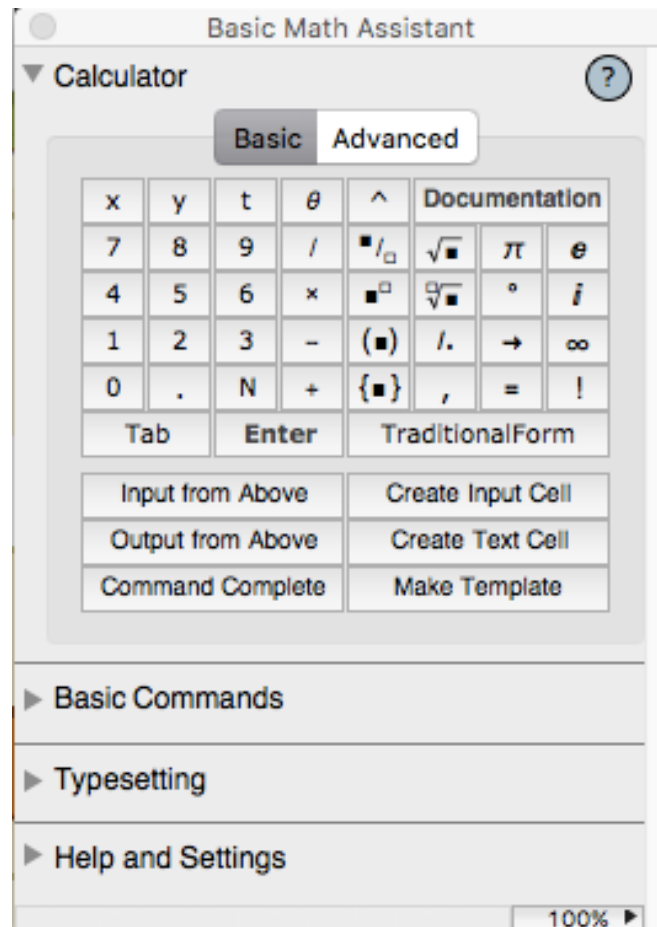
- The *GUI Palettes* → *Basic Math Assistant* helps a lot!
- Special characters are obtained with ESCAPE Commands:
 - `ESC a ESC` → α
 - `ESC p ESC` → π
 - `ESC ii ESC` → i (imaginary unit, also **I**)
 - `ESC ee ESC` → e (base of natural logarithm, also **E**)
 - `ESC inf ESC` → ∞ (infinity)
 - `ESC elem ESC` → \in (element of a set)
- Special formatting (exponent, fraction,..) is obtained with
 - `CTRL ^` or `CTRL 6` → Exponent (PC, MAC)
 - `CTRL _` → lower index
 - `CTRL SHIFT /` → Fraction
 - `CTRL 2` → Square Root
- Note: All Commands and Functions start with Uppercase (`Sin[]`, `Log[]`, `Simplify[]`,...)

Note: The special character '2' on some keyboards does NOT work for squaring!



The Math Assistant

- All formatting, symbols, expressions etc. are collected in the Math Assistant
 - Palettes->Basic Math Assistant





Aborting Calculations

- When calculations take too long:
 - **ALT** . (or ⌘ . on MAC)
 - or *Evaluation* → *Abort Evaluation*

- This sometimes does not work.
 - Try several times
 - Kill the process in the operating system

- Remark:
 - Mathematica is divided in a graphical 'Front End' and a 'Kernel' which does the calculations.
 - The kernel can use several cores or run on a remote machine!



Numbers

In[199]:=	$3 * 10 / 2$
Out[199]=	15
In[207]:=	$\frac{1}{3} + \frac{1}{4}$
Out[207]=	$\frac{7}{12}$
In[208]:=	$\frac{5}{12}$
Out[208]=	1
In[212]:=	20!
Out[212]=	2 432 902 008 176 640 000
In[209]:=	Sin[1]
Out[209]=	Sin[1]
In[210]:=	$\text{Sin}\left[\frac{\pi}{4}\right]$
Out[210]=	$\frac{1}{\sqrt{2}}$

- Normal numbers are EXACT, with *arbitrary* precision
- Fractions are kept, but simplified
- Functions are NOT evaluated numerically by default
- To get a *numerical* value, use **N[value,precision]**
- A simplified **Postfix** syntax uses the **//** operator:
 - **value // N**
- An alternative syntax is **Prefix** notation with **@**:
 - **N @ value** (or **Cos@ π , ...**)
- These two versions work for ALL functions (with 1 arg)



Complex Numbers

```
a = 3 + 4 i; b = 1 + i;
```

```
a + b
```

```
4 + 5 i
```

```
a b
```

```
-1 + 7 i
```

```
a / b
```

```
7/2 + 1/2 i
```

```
Re[a]
```

```
3
```

- Imaginary unit is **I** or **i** (entered as **ESC ii ESC**)
- Some functions with a complex number **x** are:
 - **Re[x]** or **x//Re** real part **Re[3+I] → 3**
 - **Im[x]** imaginary part **Im[3+I] → 1**
 - **Abs[x]** absolute value **Abs[3+I] → $\sqrt{10}$**
 - **Arg[x]** angle **Arg[3+I] → ArcTan[1/3]**
 - **Conjugate[x]** **3+I//Conjugate → 3-I**
 - **ComplexExpand[x]** simplifies assuming real variables
- Functions work with complex numbers:
 - **Sqrt[3+4I] → 2+I**
 - **Exp[x+I y]//ComplexExpand**
 → E^x Cos[y] + I E^x Sin[y]



Exercise 1: Numbers and Expressions

- Calculate the square root of 2 with 100 digits precision
- Can $20!$ be divided by 45 or by 46 ?
 - What is the general approach to this type of question?
 - Play with `FactorInteger [...]`. Can you guess what the output means? Try with some simple cases
- What is `Sqrt [-4]` ?
- Add, subtract and multiply the fractions $F1=4/5$; $F2=5/6$;
- Stirling's formula $\sqrt{2 \pi n} (n/e)^n$ is an approximation of the `Factorial [...]` function $n!$ ($= 1 \times 2 \times 3 \times \dots \times n$).
How large is the relative error (in Percent) for $n=10, 50, 100$?



Lists

■ Lists

```
l = {3, 4, 5}
```

```
{3, 4, 5}
```

```
{First[l], Last[l]}
```

```
{3, 5}
```

```
l[[1]]
```

```
3
```

```
l.1
```

```
50
```

```
Table[i2, {i, 1, 5}]
```

```
{1, 4, 9, 16, 25}
```

■ $L = \{a, b, c, \dots\}$ is linear list of (arbitrary!) elements

■ Picking Elements:

- The first element is `First[L]` or `L // First`
- The last Element is `Last[L]`
- The N-th Element is `L[[n]]`

Note two parenthesis! - indices start with 1!

- Append an element with `K = Append[L, elem]`

■ Generating a list (important!):

`Table[expression, {index, start, stop}]`

■ Operations on lists:

- Operators / Functions are applied to each element:

• $1 + \{1, 2\} \rightarrow \{2, 3\}$

• $\{1, 2\} + \{1, 2\} \rightarrow \{2, 4\}$

• $\text{Sin}[\{1, 2\}] \rightarrow \{\text{Sin}[1], \text{Sin}[2]\}$ 'threads over lists'



Vectors = Lists, Matrices = Lists of Lists

- $L = \{a, b, c, \dots\}$ is a linear list of (arbitrary!) elements
- Lists can be interpreted as vectors
 - Display with `MatrixForm[L]` or `L // MatrixForm`
 - Scalar multiply with `.'` operator (for example see previous page..)
- Nested lists are matrices
 - `.'` operator works
 - Multiplication works...

```
M = {{1, 2}, {3, 4}}; MatrixForm[M]
MatrixForm=

$$\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$$

M . {1, 2}
{5, 11}
```




Expressions

```
A = 3 x + 2; B = 5 y;
```

```
A B
```

$$5 (2 + 3 x) y$$

```
% / (3 x)
```

$$\frac{5 (2 + 3 x) y}{3 x}$$

```
A
```

$$2 + 3 x$$

```
Clear[A]
```

```
A
```

$$A$$

- Mathematica works with symbolic expressions
- They can be assigned to 'variables'
- Variables do **not** need to be declared
- Assignment can be cleared with `Clear[x]`
- Common Error*: 'old stuff' stored in an expression
- > Clear everything with `ClearAll["Global`*"]`
 - Note accent 'grave': ` (not ´ or ')
- Get information about an expression with `?x`
- Note that the color of an expression changes when it is defined

```
In[40]:= Clear[x, y]
```

```
In[41]:= y = 3
```

```
Out[41]= 3
```



```
In[42]:= {x, y}
```

```
Out[42]= {x, 3}
```



Direct and Delayed Assignment

- **$x=y$** **evaluates** y and assigns the result
- **$x:=y$** keeps y as an **expression**. Y is evaluated newly whenever x is used (‘delayed assignment’)

```
In[28]:= Clear[x, y]
```

```
In[29]:= y = 3
```

```
Out[29]= 3
```

```
In[30]:= x = y
```

```
Out[30]= 3
```

Now x is 3!

```
In[31]:= x
```

```
Out[31]= 3
```

```
In[32]:= y = 4
```

```
Out[32]= 4
```

x is still 3!

```
In[33]:= x
```

```
Out[33]= 3
```

```
In[34]:= Clear[x, y]
```

```
In[35]:= y = 3
```

```
Out[35]= 3
```

```
In[36]:= x := y
```

```
In[37]:= x
```

```
Out[37]= 3
```

x is y.
y is 3 (here)

```
In[38]:= y = 4
```

```
Out[38]= 4
```

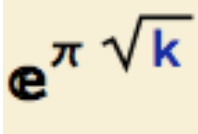
```
In[39]:= x
```

```
Out[39]= 4
```

x is y.
y is now 4



Exercise 2: Lists, Expressions

- Generate a list of the first 10 squares (n^2)
 - Pick the first, third and last element
- Define and multiply two 3-D vectors
- Define a 2×2 Matrix A. Use **MatrixForm**[...] to show it.
Let **B=Inverse**[A] (or **A//Inverse**). Multiply A and B.
- Define two expressions: $A = (3x + 5)$; $B = (5x^2 - 7)$
 - Define $F = A B$
 - Play with **Expand**[...], **Factor**[...]. Use Help to learn more.
 - Try to divide F by A. Modify F. Try **Simplify**[...]
- Calculate  for $k = 50 \dots 60$ and for $k = 160 \dots 170$
 - Look at the digits after the comma (**FractionalPart**[...])



Using Functions

- Functions & Commands have **UpperCase** names

- Arguments are given in square brackets `[]`

- `Log[x]`
- `Sin[3 x + y]`
- `FactorInteger[123456789]`

- **Important:**

Functions with 1 argument can be applied using
'**postfix**' notation

- `argument // function` `(* = function[arg] *)`
- `111 // FactorInteger` $\rightarrow \{\{3, 1\}, \{37, 1\}\}$
- `1 // Log` $\rightarrow 0$

or with '**prefix**' notation:

- `function @ argument`



Defining Functions

Important!
Common mistake!

```
x = 3;

f1[x_] = 3 + x;

f1[5]
6

f1[x_] := 3 + x;

f1[5]
8
```

Define a function with '_' after the arguments:

- `Average[x_, y_] := (x + y) / 2;`
- `Average[3, 4]` $\rightarrow 7/2$
- `DoSquare[x_] := x2;`
- `3 // DoSquare` $\rightarrow 9$

Important Detail (as discussed before):

- When defining with '=', all expressions are *immediately* evaluated
- When defining with ':=', evaluation of expressions is *delayed* until function call. This is safer, but slower!

Arguments can be constrained to a type:

```
In[64]:= g[x_Integer, y_] := x y
In[65]:= g[4, 5]
Out[65]= 20

In[66]:= g[4.5, 5]
Out[66]= g[4.5, 5]
```



Simplifying Expressions, Assumptions

```
Simplify[ $\sqrt{x^2}$ , x > 0]
```

```
x
```

```
Sin[k  $\pi$ ] // Simplify
```

```
Sin[k  $\pi$ ]
```

```
Simplify[Sin[k  $\pi$ ], k  $\in$  Integers]
```

```
0
```

```
Simplify[Cos[k  $\pi$ ], k  $\in$  Integers]
```

```
 $(-1)^k$ 
```

■ **VERY** important function: 'Simplify[...]'

• `Simplify[Sin[x]2+Cos[x]2] → 1`

or

`Sin[x]2+Cos[x]2 // Simplify → 1`

■ Often, we cannot simplify without further knowledge:

• `Sqrt[x2] // Simplify → Sqrt[x * x]`

■ We can use **Assumptions** to help:

• `Simplify[Sqrt[x2], x>0] → x`

■ Assumptions can be set globally:

• `$Assumptions = {x>0, n \in Integers, ...};`

• `$Assumptions = True; (* to clear *)`



More Powerful Simplification

- `FullSimplify[...]` makes a stronger effort

```
In[90]:= Sqrt[7 - 4 Sqrt[3]] // Simplify
```

```
Out[90]=  $\sqrt{7 - 4\sqrt{3}}$ 
```

```
In[91]:= Sqrt[7 - 4 Sqrt[3]] // FullSimplify
```

```
Out[91]=  $2 - \sqrt{3}$ 
```

Surprising ?
Do YOU understand why?

- **Very similar:** `Refine[expr, assumption]`



Manipulation Expressions

```
a = x + 5; b = 2 x + y;
```

```
a b
```

```
(5 + x) (2 x + y)
```

```
a b // Expand
```

```
10 x + 2 x2 + 5 y + x y
```

```
1 + x + y + x y // Factor
```

```
(1 + x) (1 + y)
```

```
a / b
```

$$\frac{5 + x}{2 x + y}$$

```
a / b // Expand
```

$$\frac{5}{2 x + y} + \frac{x}{2 x + y}$$

■ Many Build-In functions to manipulate Expressions:

- **Expand**
- **Factor**
- **TrigExpand**
- **TrigFactor**
- **TrigReduce**
- **ComplexExpand**
- ...

■ See help...

```
Sin[3 x] // TrigExpand
```

```
3 Cos[x]2 Sin[x] - Sin[x]3
```

```
Sin[3 x] // TrigFactor
```

```
(1 + 2 Cos[2 x]) Sin[x]
```

```
2 Sin[x] Cos[y] // TrigReduce
```

```
Sin[x - y] + Sin[x + y]
```

```
Exp[1 + I] // ComplexExpand
```

```
e Cos[1] + i e Sin[1]
```




Replacing Parts of an Expression with /.

```
Sin[x] /. x -> 3 y
```

```
Sin[3 y]
```

```
EX = a x2 + 13;
```

```
EX /. x -> (1 + y)
```

```
13 + (1 + y)2 (6 + y)
```

```
% // Expand
```

```
19 + 13 y + 8 y2 + y3
```

```
Log[t] /. t -> {1, x, Sin[x]}
```

```
{0, Log[x], Log[Sin[x]]}
```

- A symbol in an expression can be replaced by something /. expression -> newvalue

- The right hand side (expression -> newvalue) is a *transformation rule*

- **xxx /. yyy** means:

- Apply the replacement rule yyy to expression xxx
- yyy *must be* one (or a list of) transformation rules

- **Examples**

• Sin[x] /. x -> 3 y → Sin[3y]

• Log[x] /. x -> Sin[k] → Log[Sin[k]]

- Can sometimes be used in a more general way:

• (1+x)² /. (1+x) -> y → y²

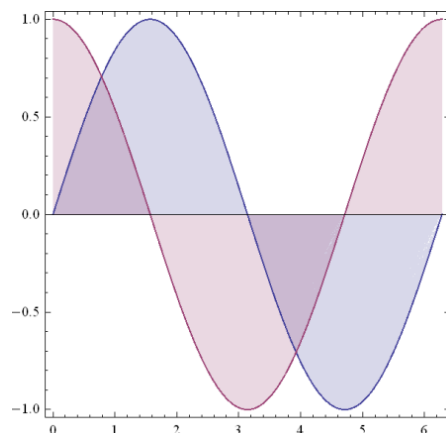
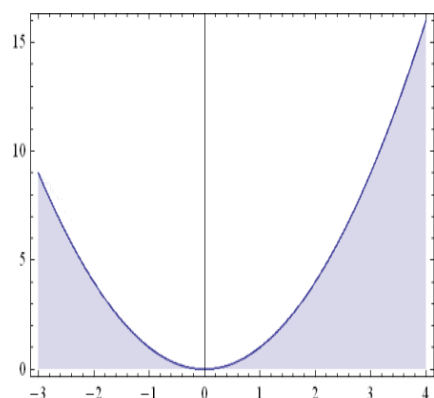
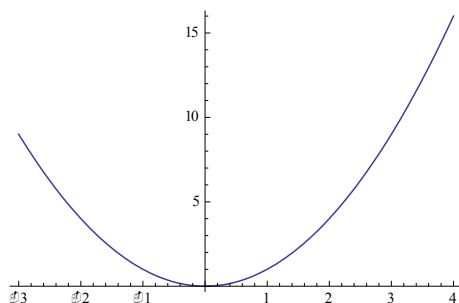


Exercise 3: Manipulations & Replacements

- Expand the expression $A = (1+3x) (1+6x)$
- Replace x by $1 + \sqrt{y}$
- Simplify the expression
- Try to factor it
- Replace y by -1 . Assign this to B
- Try `Solve[A==B, x]`. Does the result make sense?



Simple Plotting



- `Plot[expression, {var, start, stop},...parameters...]`
 - `Plot[x2, {x, -3, 4}]`
- Parameters can be used to change the display:
 - `Frame->True`
 - `Filling->Axis`
 - `ImageSize->300, AspectRatio->2`
 - `PlotRange->{ymin,ymax}` or
`PlotRange->{{xmin,xmax},{ymin,ymax}}`
 - `Plot[x2, {x, -3, 4}, Frame->True, Filling->Axis, ImageSize->300]`
- Several expressions can be plotted at once:
 - `Plot[{Sin[x], Cos[x]}, {x, 0, 2π}]`
- The options can be set globally:
 - `Options[Plot]` (* show the options *)
 - `SetOptions[Plot, {...}];` (* set them forever *)



Exercise 4: Simple Plotting

- Define function `f1[k_,x_] := Sin[k x];`

- Plot `f1` for `k=1` for `x` from 0 to 2π .
 - Play with Options, e.g.
`Frame->True, Filling->Axis, ImageSize->300`
 - Try to find some more options in the help

- Now use the list `{1,2}` for `k` to get multiple plots
 - How can you get the same result when you plot directly
`Plot[Sin(k x) /. k->{...}, {...}]`
 (Using a replacement rule for `k`)

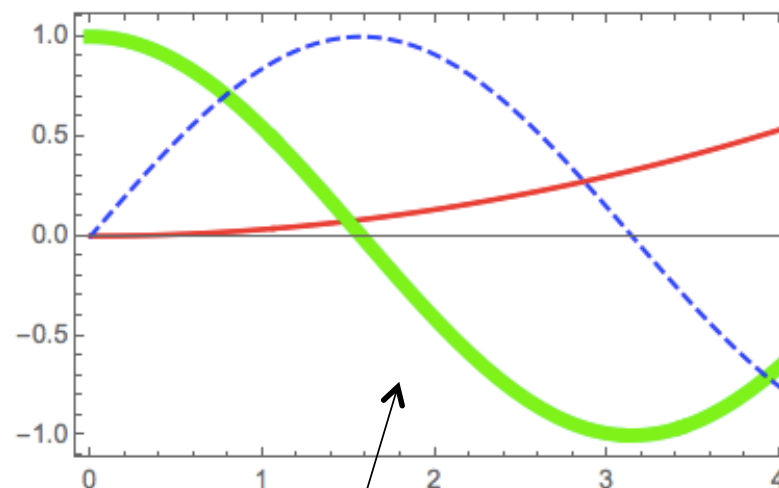
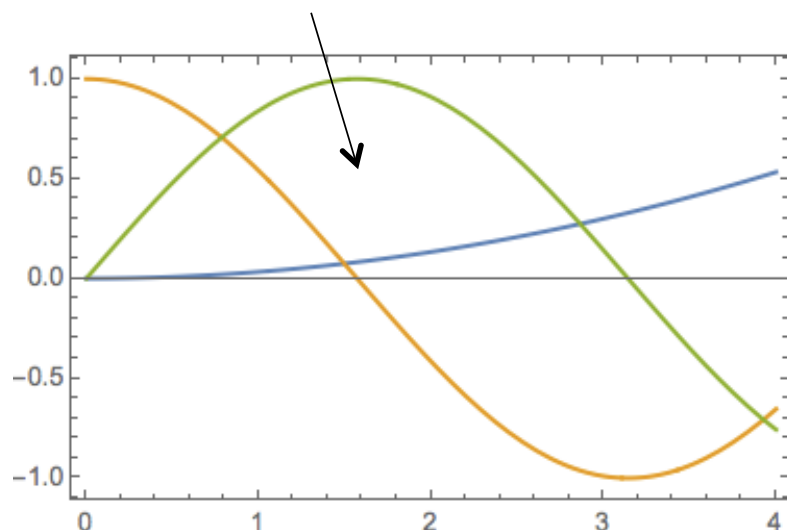
- Plot `sin(x)` and `cos(x)` in the same plot. Hint: Use a list!



(Plotting Several Curves)

- *A list of functions* automatically plots with different colours:

- `Plot[{x^2/30, Cos[x], Sin[x]}, {x,0,4}]`



- The way the individual curves are displayed can be changed with the `PlotStyle->` command. For instance:

- `Plot[{x^2/30, Cos[x], Sin[x]}, {x,0,4},
PlotStyle -> {{Thick, Red}, {Thickness[0.02],
Green}, {Dashed, Blue}}
]`



(Plotting: Why do curves look the same?)

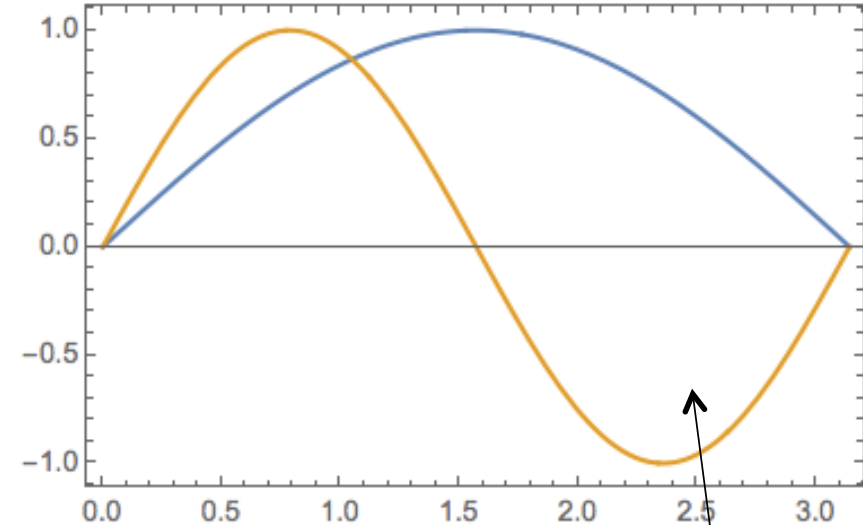
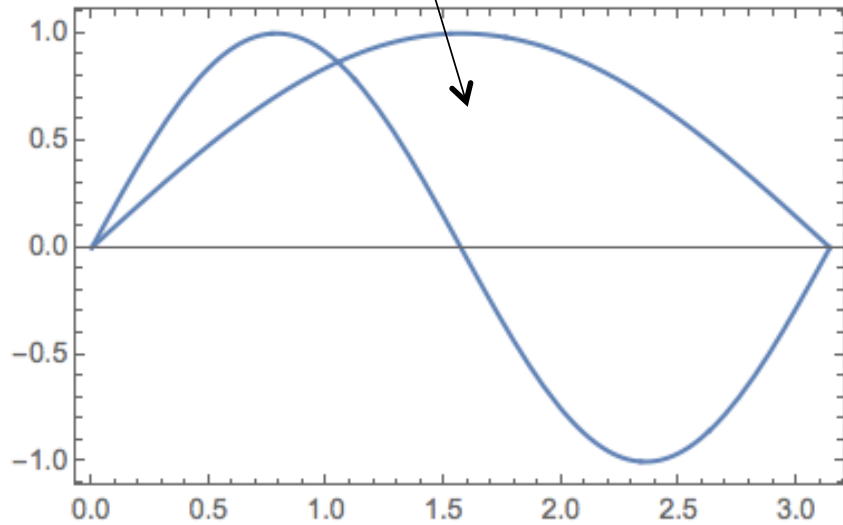
- Sometimes, curves look the same:

```
Plot[Sin[k x] /. k -> {1, 2}, {x, 0, π}]
```

or

```
Plot[Table[Sin[k x], {k, 1, 2}], {x, 0, π}]
```

gives:



- The reason is that the (function) expression is not *Evaluated* and is considered as a 'group' of curves. Evaluation can be forced (here using prefix notation):

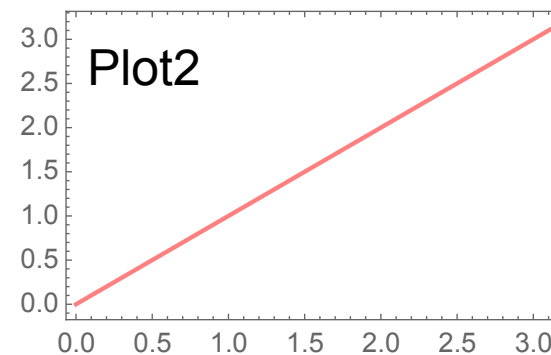
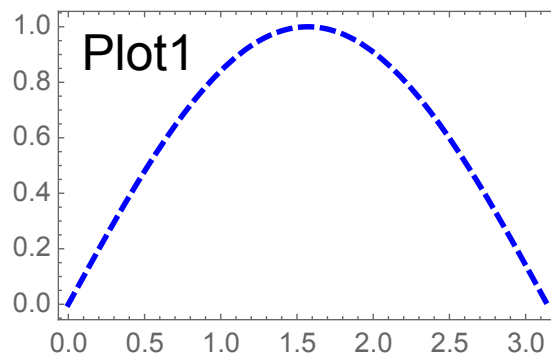
```
Plot[Evaluate@Table[Sin[k x], {k, 1, 2}], {x, 0, π}]
```



(Merging Several Plots)

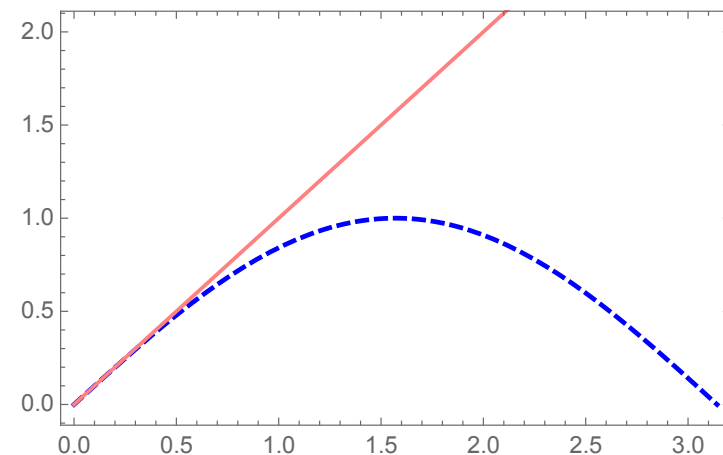
- Plots can be stored

```
Plot1 = Plot[Sin[x], {x, 0,  $\pi$ },  
  PlotStyle -> {Thick, Blue, Dashed}]  
Plot2 = Plot[x, {x, 0,  $\pi$ }, PlotStyle -> Pink]
```



and combined in an image:

```
Show[Plot1, Plot2,  
  PlotRange -> {0, 2}]
```



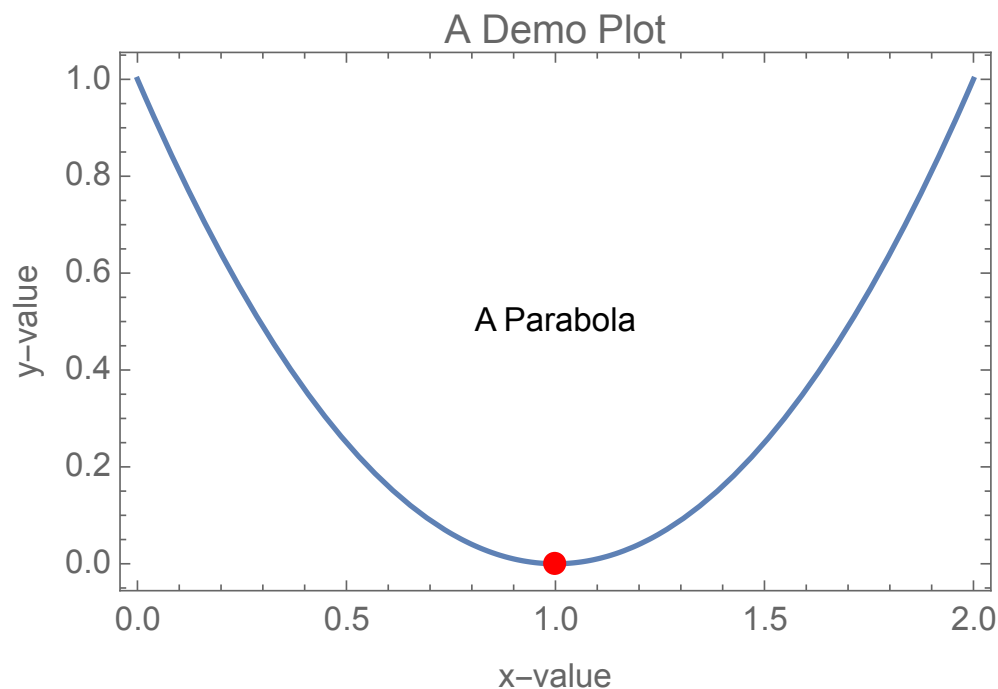


(Adding Extra Stuff to Plots)

- To add stuff, you can `Show[]` the plot with other graphics objects *or* add the 'stuff' in an `Epilog->` command:

```
Plot[(x-1)^2, {x,0,2}
,PlotLabel -> "A Demo Plot"
,FrameLabel -> {"x-value", "y-value"}
,Epilog -> {Text["A Parabola", {1, 0.5}],
Red, PointSize[Large], Point[{1, 0}]}
```

]





Solving Equations

- **Equality** in Equations must use the '==' sign
- **Solutions can be found with** `Solve[equation, variable]`:
 - `Solve[2x==4,x]` \rightarrow `{{x→2}}`
- **Solve** can find several solutions:
 - `EQ = 2 x2 == 4;`
 - `Solve[EQ,x]` \rightarrow `{{x → -√2},{x → √2}}`
 - `EQ /. %` \rightarrow `{True,True}`
- **Several equations can be solved simultaneously with '&':**
`Solve[2x+y==10 && x-y==2,{x,y}]`
- **Solve[...]** may not find all solutions. Try **Reduce[...]**:
 - `Solve [Sin[x]==0,x]` \rightarrow `{{x→0}}` (*with warning*)
 - `Reduce[Sin[x]==0,x]` \rightarrow
`C[1]∈Integers && (x==2πC[1] || x==π+2πC[1])`
- **Minima can be found with** `Minimize[f,x]`



Solving Equations

- Note: Equations must have a '==' sign or similar!!
- The result of `Solve[]` (not for `Reduce[]` ...) is always a *list* of replacement rules (even if there is only one result)
- To assign the result to an expression/variable, use `/.`:

```
EQ = 3 x + 5 y == 17;

ss = Solve[EQ, y]

{{y -> 1/5 (17 - 3 x)}}

EQ /. ss

{True}

EQ /. First[ss]

True
```

List of rules

ss solves EQ

First[%] is {y->...}

```
ysol = y /. First[%]

1/5 (17 - 3 x)

ysol

1/5 (17 - 3 x)
```

ysol now contains
the result



Reduce

- Reduce is more general. It can also solve inequalities:

```
In[215]:= Solve[2 x > 1, x]
Out[215]= {{}}
```

```
In[216]:= Reduce[2 x > 1, x]
Out[216]= x > 1/2
```

- Often the domain must be given:

```
In[211]:= Reduce[Sin[x] > 1/2, x]
Out[211]= Reduce[Sin[x] > 1/2, x]
```

```
In[212]:= Reduce[Sin[x] > 1/2, x, Reals]
Out[212]= C[1] ∈ Integers && 1/6 (π + 12 π C[1]) < x < 1/6 (5 π + 12 π C[1])
```



Some More Details

- Reduce *can* just 'simplify' expressions (without argument!)
- `Reduce[3x+6y==7 && 7x-y==4]`

$$\rightarrow y == 37/45 \ \&\& \ x == 31/45$$

- If the solution is not unique, the variables to eliminate must be specified:

- `Solve[3x+48y+5z/8==12 && x-8y-z/8==4, {x,z}]`

$$\rightarrow \{ \{x \rightarrow 4 - y, z \rightarrow -72 y\} \}$$

- `Solve[3x+48y+5z/8==12 && x-8y-z/8==4, {y,z}]`

$$\rightarrow \{ \{y \rightarrow 4 - x, z \rightarrow 72 (-4 + x)\} \}$$



Exercise 5A: Solving Equations

- 3 friends Andy, Bob, Conny have different ages:
 1. Conny is 2 years older than Andy
 2. Conny is twice as old as Bob
 3. Together they are 38 years old
- How old are they?

- Replace the first condition by
 1. Conny is older than Andy

- Can you `Solve[]` now? Try `Solve[... , {A,B}]`
- Try `Reduce[... , {A,B}]` !
- What do you learn about Conny's age?
- Add the condition that all ages are > 0 .



Exercise 5B: Solving Equations

- Define two functions $f_1=3x+2$ and $f_2=2(x-2)^2$
- Plot them ($x=0\dots5$)
- Find the intersections. Assign the result to `sol`
- Try


```
pp = {x, f1[x]} /. sol
```
- What happens?

- Add the points `Point[pp]` to your plot with `Epilog->`

- You may try $f_1=x+4$ and $f_2=3x^3 - 2x^2 - x + 5$
- (There is a problem if you replace $3x^3$ by $2x^3$)



Exercise 5C: Quadratic Equation and More

1. Quadratic Equation

- Solve the quadratic equation $a x^2 + b x + c == 0$
- Assign the solutions to two variables $x1$ and $x2$
- Check that $x1$ and $x2$ are indeed solutions

2. Try to solve a general polynomial of 3rd, 4th degree

3. Generate 10 Polynomials of the form $1+x^n$ in a list.

- Try to `Factor[...]` them. Which of them are 'prime'?

4. Generate a random polynomial of 5th order

- Generate random numbers between $-1..1$ with `RandomReal[]`
- Use `sum[...,{...}]` to generate the polynomial (use help!)
- Plot it. (For fun, plot several random polynomials in one plot)
- solve it numerically using `Nsolve[]`
- Eliminate all complex solutions with assumption $x \in \text{Reals}$



Exercise 6: Maximal Area of a Rectangle

- Which Rectangle of periphery P has largest area A ?
 - We treat a rectangle with sides a, b , periphery P and area A
- 1. Find a for given P, b (from $P == 2a + 2b$)
- 2. Inject this value into the equation for A
- 3. `Maximize[]` the area
- Is this a square ?
- Plot the area for $P=40$ as a function of b



Let's do Mathematics: Derivatives

■ Mathematica can do symbolic derivatives

- `D[expression, var]` (* first derivative *)
- `D[expression, {var,n}]` (* n-th derivative *)
- `f' [x]` (* can use ' *)

```
D[3 x^2 + 5 x, x]
```

```
5 + 6 x
```

```
D[Log[Sin[x + y]], y]
```

```
Cot[x + y]
```

```
f[x_] = (Cos[x] Sin[5 x])^2; f'[x]
```

```
10 Cos[x]^2 Cos[5 x] Sin[5 x] - 2 Cos[x] Sin[x] Sin[5 x]^2
```

```
Table[D[Sin[x], {x, n}], {n, 0, 4}]
```

```
{Sin[x], Cos[x], -Sin[x], -Cos[x], Sin[x]}
```

```
D[Sin[x]^10, {x, 4}]
```

```
5040 Cos[x]^4 Sin[x]^6 - 4680 Cos[x]^2 Sin[x]^8 + 280 Sin[x]^10
```



Integrals

■ Indefinite Integrals

- `Integrate[expression, var]`

■ Definite Integrals

- `Integrate[expression, {var, start, stop}]`

```
Integrate[Log[x], x]
```

```
-x + x Log[x]
```

```
Integrate[1/(x^3 + 1), x] // Simplify
```

$$\frac{1}{6} \left(2\sqrt{3} \operatorname{ArcTan}\left[\frac{-1+2x}{\sqrt{3}}\right] + 2\operatorname{Log}[1+x] - \operatorname{Log}[1-x+x^2] \right)$$

```
Integrate[Exp[-c x^2], {x, -∞, ∞}]
```

$$\operatorname{ConditionalExpression}\left[\frac{\sqrt{\pi}}{\sqrt{c}}, \operatorname{Re}[c] > 0\right]$$

```
{∫₀^π Sin[x] dx, Integrate[Cos[x], {x, 0, π}]}
```

```
{2, 0}
```



Other Analysis Stuff

■ Limits (also for $x \rightarrow \infty$)

- `Limit[Sin[x]/x, x → 0]` → 1

■ Sums

- `Sum[k2, {k, 1, n}]` → $\frac{1}{6} n (1+n) (1+2 n)$

- `Sum[1/k2, {k, 1, ∞}]` → $\pi^2/4$

$$\text{Limit}\left[\frac{\text{Sin}[x]}{x}, x \rightarrow 0\right]$$

1

$$\text{Limit}\left[\frac{\text{Sin}[x + \epsilon] - \text{Sin}[x]}{\epsilon}, \epsilon \rightarrow 0\right]$$

`Cos[x]`

$$\text{ArcTan}[\infty]$$

$\frac{\pi}{2}$

$$\text{Limit}\left[\frac{\text{Cosh}[x]}{\text{Sinh}[x]}, x \rightarrow \infty\right]$$

1

$$\text{Sum}[k^2, \{k, 1, n\}]$$

$$\frac{1}{6} n (1+n) (1+2 n)$$

$$\text{Sum}[\text{Table}[1/k^n, \{n, 1, 4\}], \{k, 1, \infty\}]$$

`Sum::div` : Sum does not converge. \gg

$$\left\{ \sum_{k=1}^{\infty} \frac{1}{k}, \frac{\pi^2}{6}, \text{Zeta}[3], \frac{\pi^4}{90} \right\}$$

$$\text{Sum}\left[\frac{x^n}{n!}, \{n, 0, \infty\}\right]$$

e^x



Exercise 7: Area of a circle

- Write down the relation EQ between x , y , r for a circle
 - Solve this for y (at given x , r) and define a function $y[x, r]$
 - Plot a half circle (x from $-r$ to r)
 - Make sure the plot is properly scaled
 - Calculate the area by integration from $-r$ to r (this takes time..)
 - Simplify the result by using $r > 0$
-
-
-
- Now first calculate the general integral 'Integral $y[x, r]$ dx '
 - Plot how the area increases as x increases from -1 to 1 (for $r=1$)
 - Try to set $r=x$.
Get rid of the indefinite expression with `Limit[...]`
 - Is the result as expected?
 - Use the option `Direction` in `Limit[...]`



Exercise 8: Some More Plotting & Stuff

- Define the function $f[x_] = \sin[a x] \exp[-\lambda x]$
 - Use Assumptions $a > 0$ and $\lambda > 0$
- Try Solve/Reduce to find a such that the first zero is at $x=1$...
 - Also try to prove that the a you chose leads to $f[x]=0$ for integer value of x
- Plot the function with this a for some values of λ
- You may know that a saw tooth Function can be decomposed (Fourier transformed) into functions $\sin[kx]$ with coefficients $(-1)^k / k$
 - Define the k -th such component as $g[x, k]$
 - Plot these base functions with increasing k (use `Table[...]`)
 - Define the sum of the first k components $f[x, k]$ (use `Sum[...]`)
 - Plot some sums with increasing number of overtones to show how the saw tooth function is approximated

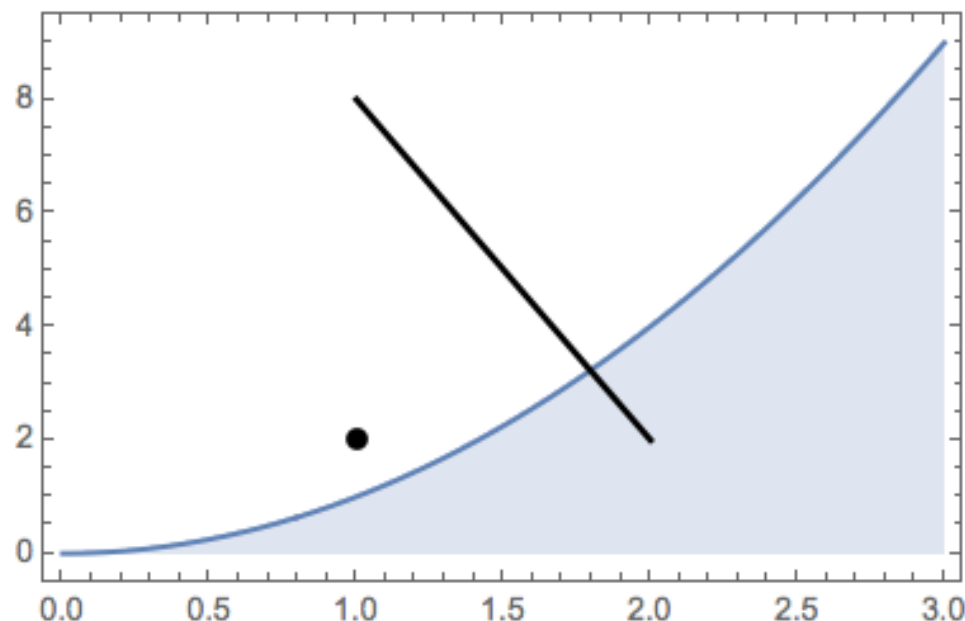


ADVANCED TOPICS



Graphics

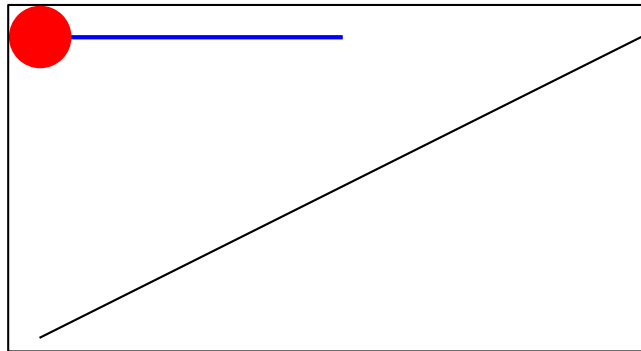
- Result of `Plot[...]` is a *graphic* object:
 - `Plot1 = Plot[x2, {x, 0, 3}];`
- There are other graphic objects:
 - `APOINT = Graphics[Point[{1, 2}]];`
 - `ALINE = Graphics[Line[{2, 2}, {1, 8}]];`
- They can be plotted together:
 - `Show[Plot1, APOINT, ALINE];`



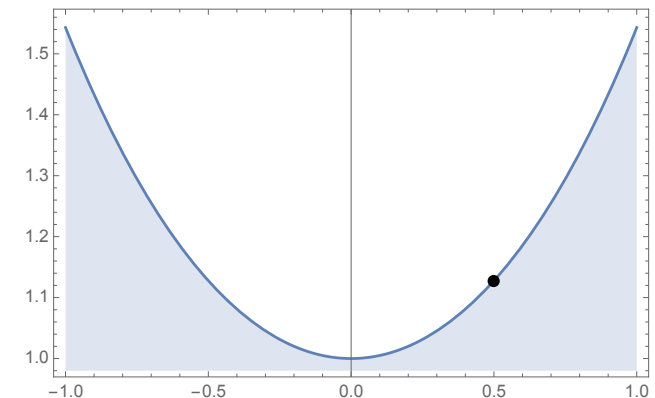


Graphics

- Graphics objects can be merged in list (with `{ .. }`):
 - `Graphics[{Line[{{1, 1}, {3, 2}}], Thick, Blue, Line[{{2, 2}, {1, 2}}], PointSize[0.1], Red, Point[{1, 2}]}]`



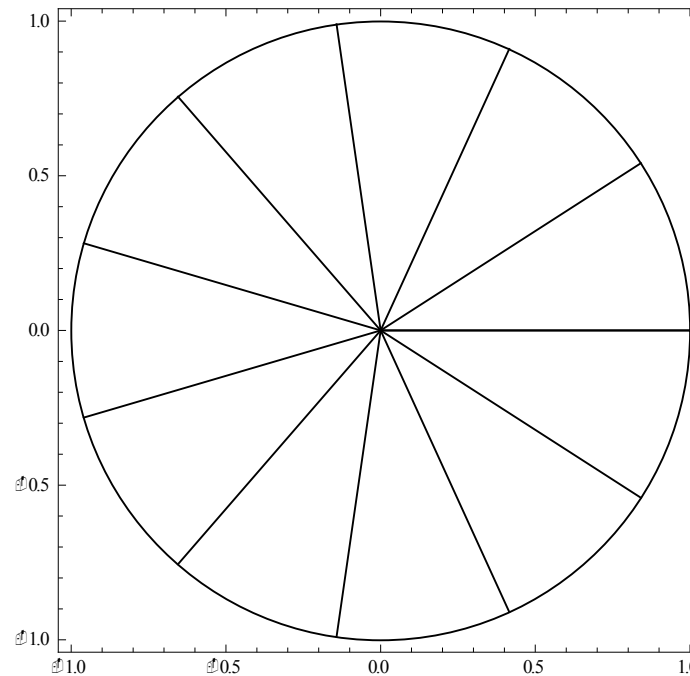
- To directly include graphics in a `Plot[]`, you can use `Epilog`:
 - `Plot[Cosh[x], {x, -1, 1}, Epilog -> {PointSize[Large], Point[{0.5, Cosh[0.5]}]}]`





Exercise 9: Drawing a Wheel

- Draw the following Wheel:

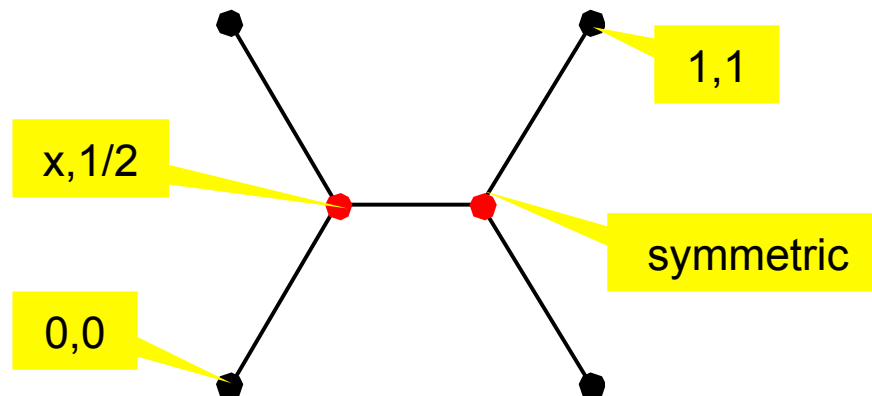


- Use the help function
- Hint: you get one spoke by
`Line[{{0,0}, {Cos[2 π α], Sin[2 π α]}}]`
- Make a `Table[]` of such spokes for α from 0 to 1 in 11 steps
- `Show[Graphics[{...the spokes..., ...the circle...}]]`



Exercise 10: Minimizing a Mesh

- We want to connect the two red points to four corners with minimal total distance.
 - The corners are at $x/y = 0/1$, the left dot at $x/0.5$



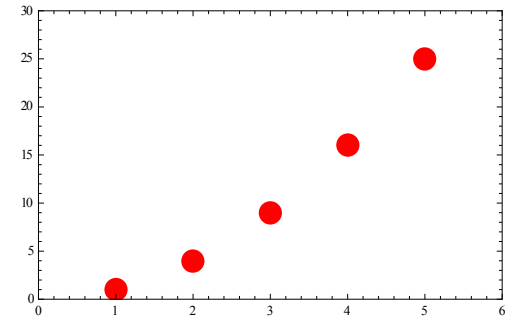
- Calculate the total distance as a function of x
- Plot it for $x = 0 \dots 0.5$
- For which x is it minimal ? Plot that point into the diagram!
- At which angles do the lines join at the red points? (half angle?)
- Define 4 corner points and a list `corners=Point[{p0,p1,p2,p3}]`;
- Define a graphic which shows the above figure.



More 2D Plotting

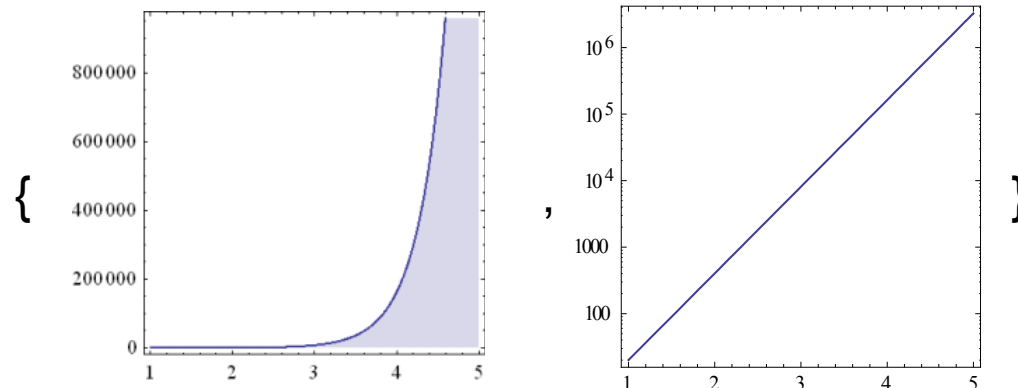
■ Plot data sets (lists): `ListPlot[list]`

- `ListPlot[Table[i2, {i, 1, 5}],
PlotRange->{{0, 6}, {0, 30}},
PlotStyle->{PointSize[0.05], Red}]`



■ Axis scaling: `LogPlot[...]`, `LogLogPlot[...]`

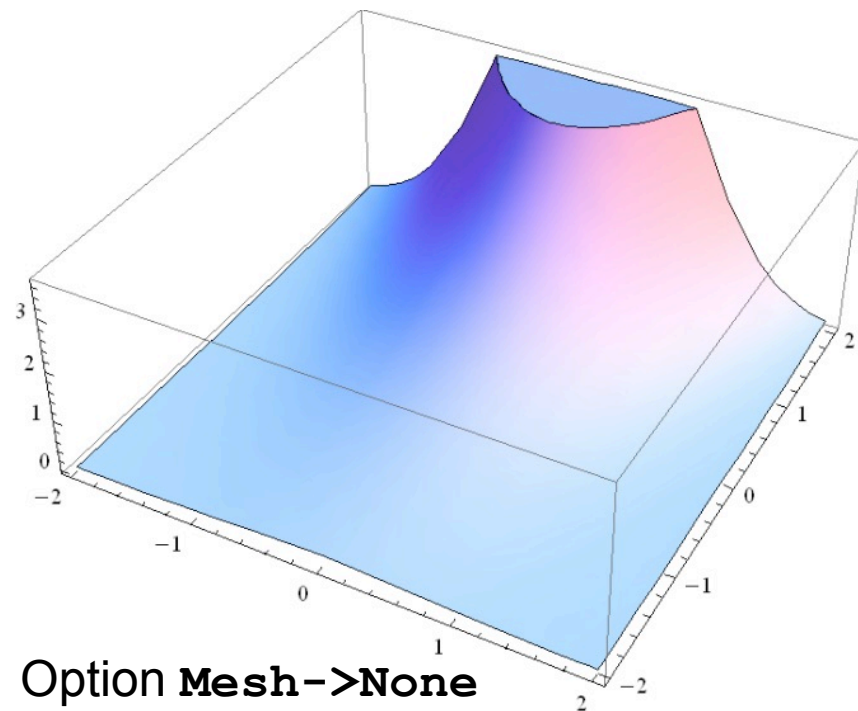
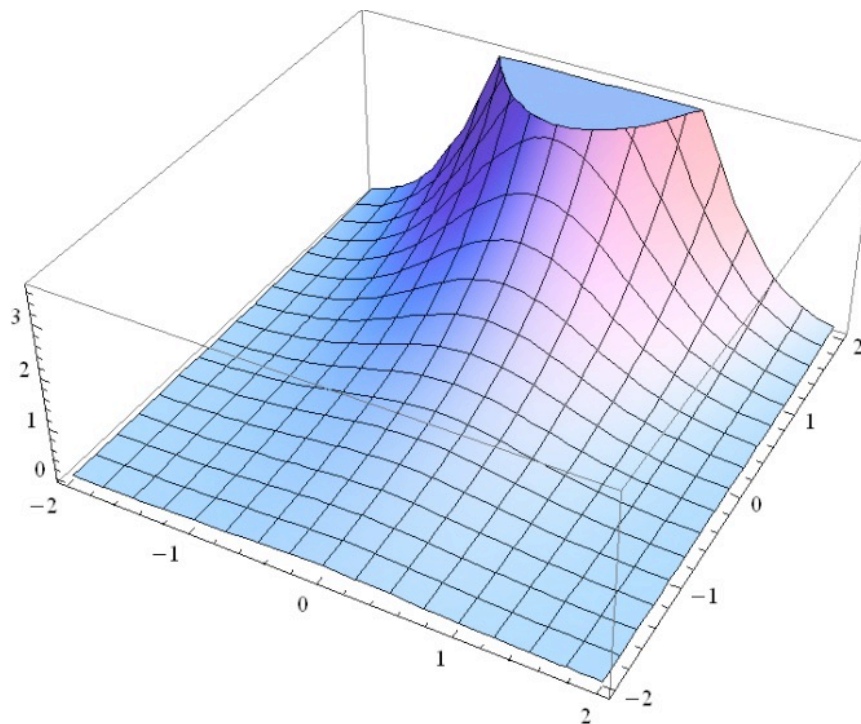
- `{Plot[Exp[3x], {x, 0, 5},
ImageSize->200, Frame->True, AspectRatio->1],
LogPlot[Exp[3x], {x, 0, 5},
ImageSize->200, Frame->True, AspectRatio->1]}`





3D Graphs

- Many ways to show surfaces, ...
- For instance
 - `Plot3D[f[x,y], {x,-2,2}, {y,-2,2}, ...options...]`



- View position can be changed with the mouse.
- Many options (see help)



Data & Image File IO

■ Can save plot results to file:

- `SetDirectory["C:/..."]`
or
`SetDirectory[$HomeDirectory <> "/xxx/xxx"]`
or
`SetDirectory[NotebookDirectory[] <> "subdir"]`
- `PLOT = ...`
- `Export["file.pdf", PLOT] (* or .jpg, .gif, ... *)`

■ Can also read data from file:

- `A = Import["file"];`

■ You can specify data types, skip lines, ...:

- `Import["filename", "csv", HeaderLines -> 1];`



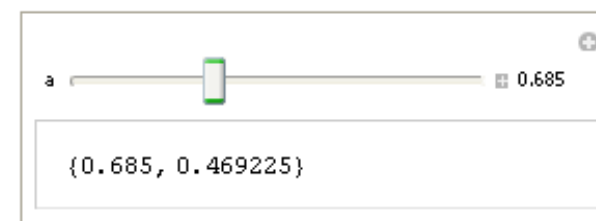
Manipulate: Direct User Interaction

- Manipulate is used for interactive animations:

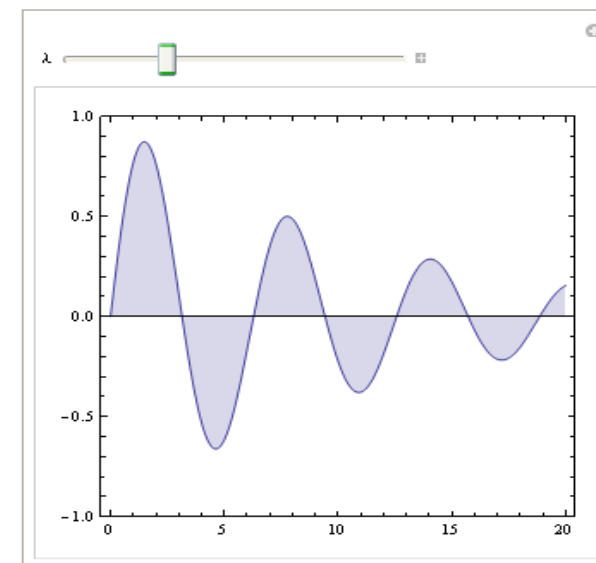
- Manipulate[
 - ... some commands ...
 - { {variable,default,"caption"}, min, max,
 - Appearance->"Labeled" }

- For instance:

- Manipulate[
 - {a,a²},
 - {{a,1,"a"},0,2,
 - Appearance->"Labeled" }



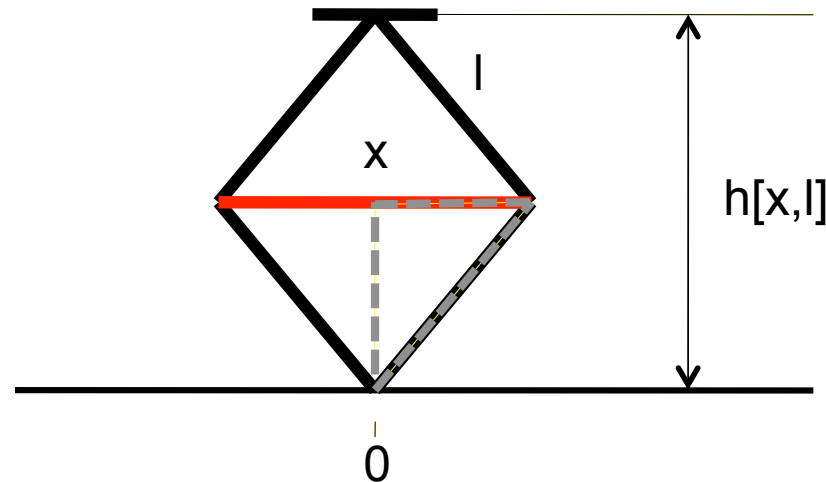
- Manipulate[
 - Plot[Exp[-λ x] Sin[x],{x,0,20},
 - PlotRange->{-1,1}],
 - {λ,0,0.3}





Exercise 11: Car Lifter

- A car lifter is composed of a (black) frame with 4 edges of length l and of a (red) drive of variable length x (see figure).



- Calculate the height h as a function of x . (You can solve Pythagoras's equation for the gray triangle for x)
- Plot this (what is the range of x ?)
- Plot the vertical force created by a constant force on x as a function of x (hint: you need a derivative!)
- If you want: Make a graphics and try Manipulate...



What Else?

- Differential Equations (analytical, numerical)

```
DSolve[f'[x]2 == f[x], f[x], x]
```

```
{ { f[x] →  $\frac{1}{4} (x^2 - 2 x C[1] + C[1]^2)$  }, { f[x] →  $\frac{1}{4} (x^2 + 2 x C[1] + C[1]^2)$  } }
```

```
DSolve[Sin[x] f'[x] == f[x], f[x], x]
```

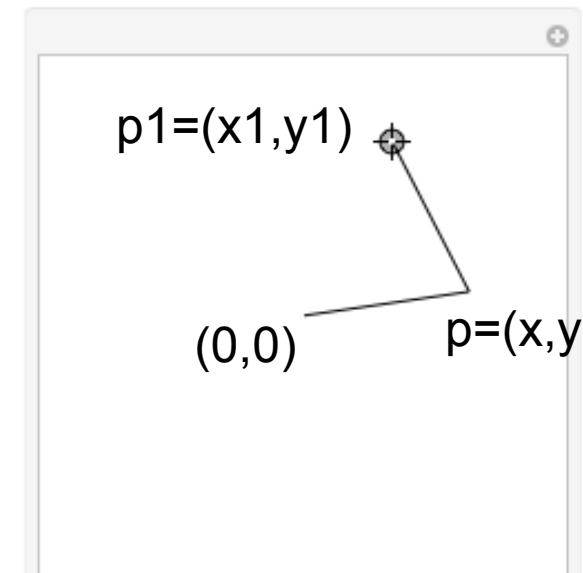
```
{ { f[x] →  $C[1] \tan\left[\frac{x}{2}\right]$  } }
```

- Dynamical Variables
- Many packages
-



Exercise 12: Equal Trace Lengths

- On a 2D detector, we want to connect an amplifier at $(x,y)=0$ to a pixel at a position $p1=(x1,y1)$.
- We want to keep the trace length constant (same capacitance)
- For simplicity, we use 2 straight lines of equal length with bend at $p=(x,y)$
- Limit all coordinates $(x,y,x1,y1)$ to >0
- Define points $p1$ and p
- Calculate the length of the 2 straight parts as their `Norm[]`
- Find (x,y) such that both parts have equal length and the sum is some constant $K>0$.
- Assign the result to a point `result[x,y,K]`
- Try some special cases. Plot.
- Use Manipulate with a 'Locator' (see help!)





Common Mistakes

- Expression has already been assigned
 - Observe the display colour!
 - Check with `?Expression`
 - Clear with `Clear[Expression]`

- Variables are not separated in the input:
 - `y = ax` is different from `y = a x`

- Mismatch in number of arguments:
 - `A=Sin[x]; Plot[A[x], {x, 0, 3}]` -> no result
 - `A[x_]=Sin[x]; Plot[A, {x, 0, 3}]` -> no result

- Subtle: Assignment contains command

```
In[35]:= x = {1, 2} // MatrixForm
```

```
Out[35]//MatrixForm=
```

$$\begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

```
In[36]:= 3 x
```

```
Out[36]= 3  $\begin{pmatrix} 1 \\ 2 \end{pmatrix}$ 
```



Homework1: Low Pass & Notch Filter

- The complex impedance of a capacitor is $1/(i \omega C)$
- Write down the transfer function (v_{OUT}/v_{IN}) of a RC low pass
- Determine the absolute value of the output voltage
 - Use `Abs[...]`
 - You must use appropriate `$Assumptions` to simplify!
 - Use `FullSimplify[...]`
- Plot the transfer function for $RC=1$ from $\omega=0.01\dots 100$
 - Linear
 - Double log using `LogLogPlot[...]`
- Connect
 - A) **two** high pass sections in series
 - B) **two** low pass sections in series
 - Add the voltages
- When is the output amplitude zero? Plot the result!



Homework 2: 'Special'

- Since some years, the DPG research centre MATHEON publishes a mathematical 'Adventskalender'
- Solve the following exercise (from <http://www.mathekalender.de>)

Aufgabe:

Beim Einpacken von Weinflaschen in viereckige Geschenkkartons fällt dem Weihnachtsmann auf, dass er jedesmal eine Menge Luft mit einpackt. Und außerdem verbrauchen diese Ecken sicher viel zu viel vom teuren Karton. „Wenn ich die Ecken abschneide und durch eine gerade Verbindung ersetze, kann ich die Verpackung aus weniger Karton falten“, überlegt er. Flugs macht er sich ans Werk und umhüllt die nächste Flasche mit einem achteckigen Karton. Zufrieden packt er weiter, bis ihm nach einer Weile auffällt, dass immer noch zu viel Karton verbraucht wird. „Wenn ich die Ecken wieder abschneide, sollte ich noch weniger Karton nehmen können“, sagt er sich und falzt eine sechzehneckige Verpackung. Nun packt ihn der Ehrgeiz. Schnell ist ihm klar, dass kreisförmige (zylindrische) Verpackungen optimal wären, nur leider lässt sich der Karton nicht gut biegen. Aber immerhin kann er mehr und mehr Ecken falzen. Er hört mit der Eckenverdoppelung erst auf, als sein Verpackungsumfang um nicht mehr als ein Tausendstel länger ist als bei einer zylindrischen Umhüllung.

Wie viele Ecken hat seine Verpackung dann?

- Define a Function $L[N]$ which gives the length of the regular polygon with N corners and (inner) radius 1. Plot the function
- Check when the function reaches $1.001 \times 2 \pi$
- Find the numerical solution with **FindRoot** (not **NSolve**)



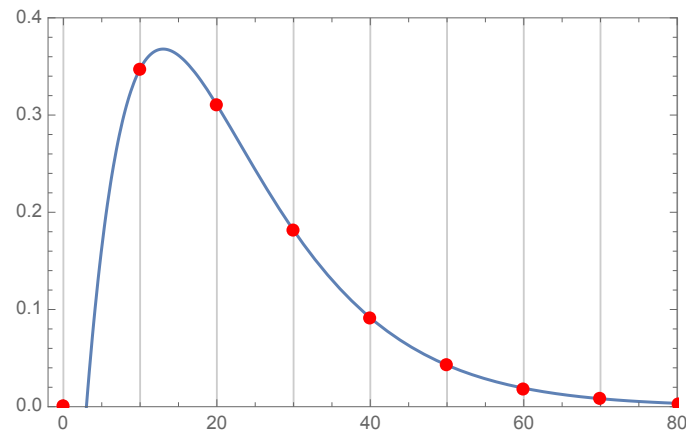
Homework 3

- Particles penetrating a material are absorbed according to an exponential extinction law $f(x) = A \text{Exp}[-\alpha x]$
- Find A so that the total number of absorbed particles is normalized to 1
- Our absorber has a total thickness T .
- We want to divide it in two layers $(0..T_1, T_1..T)$ so that an equal amount of particles are absorbed in both layers.
- Calculate T_1 as a function of α (and T)
- Plot your result
- What T_1 do you find for $T=1$ and $\alpha=1$?



Homework 4: A Real World Problem

- A detector delivers pulses with pulse shape $f(t) = t/\tau e^{-t/\tau}$ at *random* times
- An ADC samples this signal at times $0, \Delta, 2\Delta, \dots$



- We want to know the peak amplitude, but it is mostly measured too small by the random sampling.
- We want to know the average error (for all possible sampling times) as a function of the pulse time constant τ (for a given $\Delta=1$)



Homework 4, cont.

- Define the function $f[t, \tau, A]$ (A is an amplitude factor)
- Plot it for $\tau = 10$ and $A = 1$
- Find out at which time the peak is located (set the derivative to zero and solve)
- What is the peak amplitude? Find A such that the peak amp. is 1.
- Define a new function $f1[t, \tau]$ which has peak amplitude 1
- Consider the samples around the peak. One sample shall be at a time x behind the peak, the other is Δ before. Show the amplitudes of the two samples in one plot (for x ranging from $0 \dots \Delta$). Use for instance $\tau=8$ and $\Delta=10$.
- Where (at which x) do the two curves cross?
- We obviously take the larger sample as the amplitude estimate. We want to know the average amplitude (averaging over all x): Integrate the first part up to the crossing and the second part up to Δ and divide by Δ .
- What is the result for $\tau=8$?

