Lecture 10: An overview of MATHEMATICA

A8B17CAS

Jozef Lukáč

Department of Radio Engineering
Czech Technical University in Prague
Czech Republic
lukacjo1@fel.cvut.cz

November 25
Winter semester 2024/25

Outline

Wolfram Mathematica is...

- A system for technical computing.
- A set of definitions and rules for manipulation/simplification of expressions.
- Philosophy: kernel & front end (notebook[1]).
- Interpreted language (not compiled).
- Multi-licensed for CTU.
  - Available for students as well!
    - with Wolfram ID: https://ist.cvut.cz/nase-sluzby/mathematica/
    - download.cvut.cz

[1] https://en.wikipedia.org/wiki/Notebook_interface

Wolfram Mathematica vs. Maple

- Heavily used in academics.
- Helps you to solve complex problems in seconds.
- Better for (symbolic) integration.
- ▶ Why is Mathematica better.

- It has a wide range of special mathematical libraries.
- Better for ODE (ordinary differential equations), PDE (partial differential equations), and recurrence equations.
- ▶ Why is Maple better.

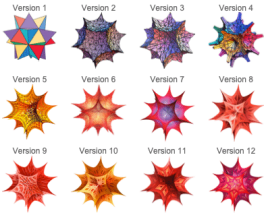
- Other differences:
  - https://www.geeksforgeeks.org/difference-between-maple-and-mathematica/
  - https://qr.ae/pvxO3O
  - https://www.reddit.com/r/Physics/comments/7gdx4g/comment/dqic7eq/?context=3

Historical Development of Mathematica

- 1986-1988
  - Stephen Wolfram[2] works on a new algebraic system – SMP (a symbolic manipulation program)
- June 23, 1988
  - Mathematica 1.0 is launched.
- current version of Mathematica is 13.1 (November 2022)
- related references
  - History of the language in pictures https://www.wolfram.com/mathematica/scrapbook/
  - Wolfram Language https://en.wikipedia.org/wiki/Wolfram_Language
  - Wiki https://en.wikipedia.org/wiki/Wolfram_Mathematica

[2] https://en.wikipedia.org/wiki/Stephen_Wolfram
https://www.stephenwolfram.com/

Wolfram Mathematica (WM) logo



History of WM logo.

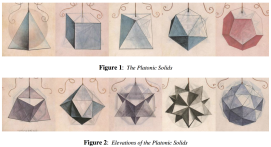


Figure 1: *The Platonic Solids*

Figure 2: *Elevations of the Platonic Solids*

Da Vinci's stellated platonic solid.

- Related references:
  - Platonic solids `https://en.wikipedia.org/wiki/Platonic_solid`
  - Spikey `https://mathworld.wolfram.com/Spikey.html`

Launching MATHEMATICA

- Desktop icon
- Command line:
  - `math`, `wolfram` or `wolframscript`

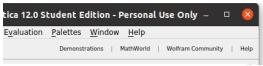
```
>wolframscript
Wolfram Language 12.0.0 Engine for Linux x86 (64-bit)
Copyright 1988-2019 Wolfram Research, Inc.

In[1]:= 5+3*2.1^2

Out[1]= 18.23

In[2]:= f1[x_]:=x^2;

In[3]:= Table[f1[n],{n,4}]

Out[3]= {1, 4, 9, 16}

In[4]:= f1/@{-1.1,3,Pi,4/5}

                      2  16
Out[4]= {1.21, 9, Pi , --}
                        25

In[5]:= Exit[]
>
```

- A user can launch MATHEMATICA with optional parameters.
  - `wolframscript -h` (cmd help)
- System requirements[3] for MATHEMATICA 13.1:
  - Windows 10+, Linux, macOS 10.14+
  - 4+ GB RAM
  - 19 GB of HDD
  - Any Intel or AMD x86-64 processor

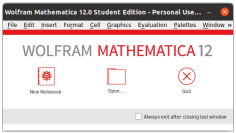
[3] https://www.wolfram.com/mathematica/system-requirements.html

MATHEMATICA termination + termination of a (long-running) computation

- Close notebook: Close button in the top right corner of the notebook window.

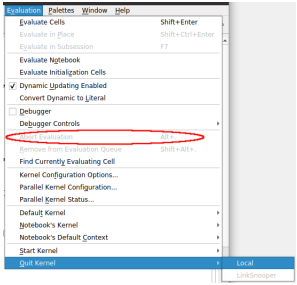


- Quit MATHEMATICA: "Quit" button in the welcome window.



- Possibility to terminate MATHEMATICA in the command window (Ctrl+c).
- When the computation runs too long ...

- ... abort evaluation (Alt+.)
- ... or quit MATHEMATICA kernel.



- Other options, see documentation[a]

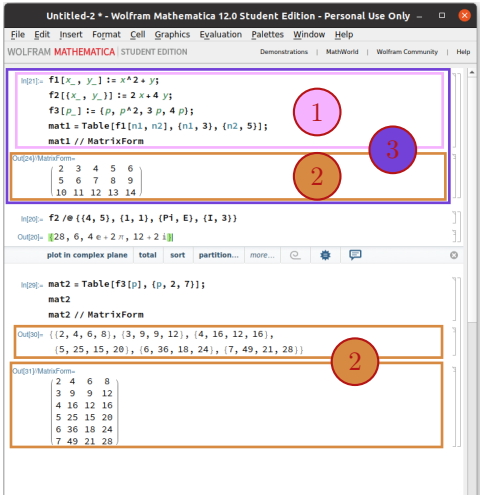
[a] https://reference.wolfram.com/language/howto/StopAComputation.html

The Mathematica's Notebook Environment

1. Input cell. (evaluate it by `Shift + Enter`)
2. Output cell(s).
3. Group of 1 input and output cell(s).

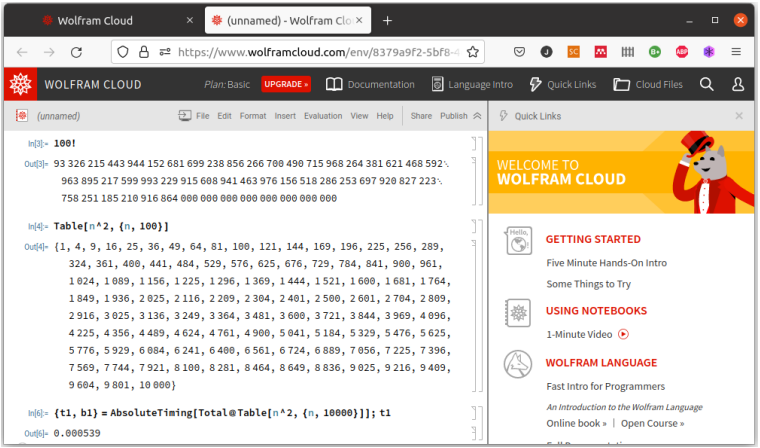
Notes:

- Input and Output cells are inner cells that are grouped in a group.
- (mouse-) double click on an inner cell closes all but this inner cell in the current group.

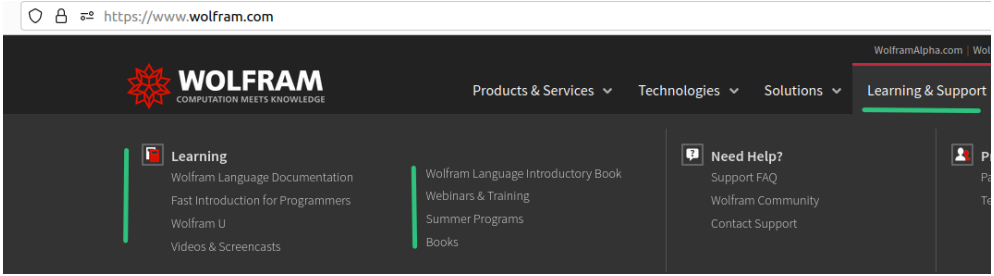


Mathematica Online

- https://www.wolfram.com/mathematica/online/
- Runs in a web browser.
- Requires (CTU) login.
- Slower than regular Mathematica.

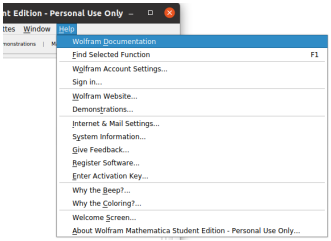


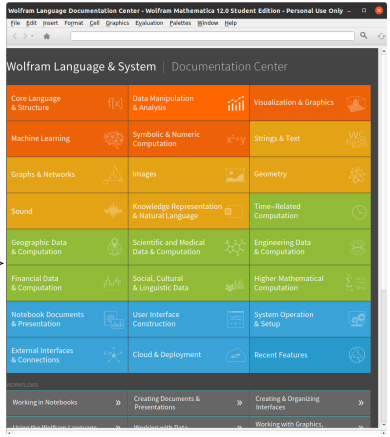
Learning from online sources (Courses, Classes, Videos, . . .)



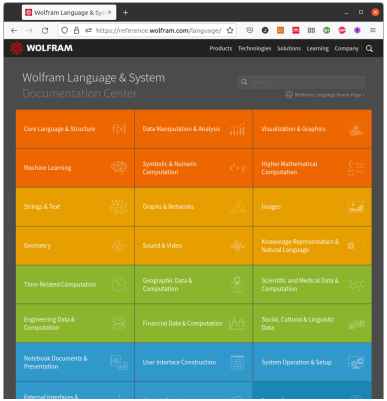
- A lot of video courses are available online at https://www.wolfram.com/ *Learning & Support* section.
- https://www.wolfram.com/language/fast-introduction-for-programmers/en/
- https://www.wolfram.com/wolfram-u/

Documentation



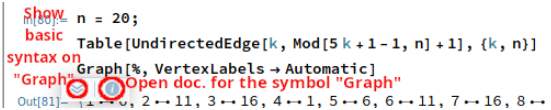


https://reference.wolfram.com/language/

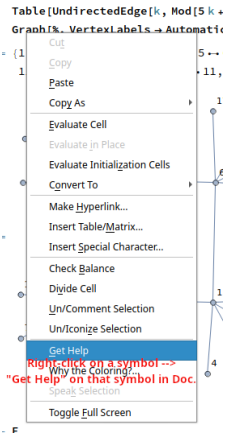


Getting Help on a Symbol

- Move mouse over a symbol, two icons will appear.



- Right-click on a symbol, and select "Get Help".



WOLFRAM language

WOLFRAM language is case-sensitive!

- As the core element of MATLAB is a matrix, the core element of MATHEMATICA is an *expression*.
  - Every expression consists of its head (e.g. `f`, `Set`, `Plus`, `Equal`,...) and arguments.
  - Examples: `f[1,5,{5,4}]`, `Set[a,5^3]` (or `a=5^3`), `Plus[5,3,9]` (or `5+3+9`), `Equal[5-3, a + b]` (or `5-3 == a + b`).
- All built-in commands start with an *upper case* letter (*e.g.* `Sin[0.4]`).
- To pass arguments to functions, use *square brackets* (*e.g.* `BesselI[0,2.0]`).
- For user-defined symbols/names use *lower case* starting letter (*e.g.* `myExpression=3+x^2-Log[y]`).
- Do not use underscore ("_") in user-defined names (*e.g.* `my_expression` – invalid). "_" is a pattern that matches one arbitrary expression (*e.g.* `f1[x_]:=Cos[x]^2`).
  - Letters and numbers are allowed; other symbols (colon ":", hyphen "-" and others) are not.
- Beware of different syntax in MATLAB, *e.g.*, `sin(x)` vs. `Sin[x]`, etc.

Variable Names

- Examples of valid symbols/names:

```
a, aA, b, c, x1, x2, m12, test1, matrixA, fx, fX
```

- Examples of invalid variable names:

```
9var          (*starts with a number*)
matrix_A     (*name cannot contain underscore*)
Test-1        (*algebraic expression*)
```

- Examples of valid numbers in Mathematica,

```
3, -66, +0.0015, .015, 1*10^2, 1.6025 10^(-10),1.6025*^-10
05.1, Pi , E , I , Infinity , 2/3, 1.2, 1.2`20, 1.2``20
```

Workspace – Output Deletion

- To delete a cell:
  - Select the cell (left-click on it) and hit the `Delete` key on the keyboard. →
- To clean one (or more) variable(s):
  - **`Clear[mySymbol1, myVariable2, myFunc3]`**
    or instead of **`Clear`** use **`ClearAll`** (removes also **`Attributes`** of the given objects).
  - Or unassign the given symbol:
    `myVar =.`

Numerical objects in MATHEMATICA

- Types of numbers in MATHEMATICA:

  Integer – arbitrary-length exact integer (*e.g.* `1000031`).
  Rational – *Integer/Integer* in lowest terms (*e.g.* `3/556`).
  Real – approximate real number with any specified precision (*e.g.* `12.`, `-3.83``15`).
  Complex – complex number of the form *number* + *number* `I` (*e.g.* `83 + 0.500 I`).

  Get the type by `Head[`*number*`]`, *e.g.* `Head[123]`, `Head[123.00]`, `Head[3/6]`.

- Objects in MATHEMATICA can be grouped by a **List**,
  (*e.g.*`{2, 3.05, 4/5, Pi^2, {2, {3}}, E^Pi}`). Elements of lists can be other lists.
- Common numeric objects:
  - scalar: a number (Integer, Rational, Real, or Complex).
  - vector: a list of scalars (*e.g.* `{5, 3.4, -6, -0.1 +3 I}`). 1D numeric object.
  - matrix: a list of (row-) vectors (*e.g.* `{{1,2},{3,4}}`). 2D numeric object.
  - array (multidimensional matrices): A list of lower-dimensional (numerical) objects.
- Matrices (resp. lists of lists) can be complex, and can contain text/strings as well.

Matrix/List Creation

- Following techniques are available:
  - element-by-element entering (suitable for small matrices/lists only),
  - concatenation of parts to one object (*e.g.* row-vectors to one matrix) – make **`List`** of them,
  - by a formula for each element – use **`Table[]`** command,
  - generation by built-in functions (**`ToeplitzMatrix[]`**, **`IdentityMatrix[]`**, **`DiagonalMatrix[]`**, **`Range[]`**),
  - **`Import`** and **`Export`** from/to external files (`.mat`, `.txt`, `.xls`, ...).

Matrix Construction Element-by-element I.

- Test the following commands to construct matrices by element enumeration.

```
a1=a2=-1;
a1
a2
v1={-1,0,1};
v2={{-1},{0},{1}}; (* !! *)
v1
v1//MatrixForm
v2//MatrixForm
matM1={{-1,0,1},{-2,0,2}};
matM2={{-1,-2},{0,0},{1,2}};
matM3={{-1,-2},{0,0}};
MatrixForm /@ {matM1,matM2,matM3}
```

- Suitable for small matrices only.

$$a_1 = a_2 = -1$$

$$\mathbf{v}_1 = \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\mathbf{v}_2 = \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

$$\mathbf{M}_1 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \end{bmatrix}$$

$$\mathbf{M}_2 = \begin{bmatrix} -1 & -2 \\ 0 & 0 \\ 1 & 2 \end{bmatrix}$$

$$\mathbf{M}_3 = \begin{bmatrix} -1 & -2 \\ 0 & 0 \end{bmatrix}$$

Notes on the syntax.

- The expression `f[expr1]` can be written in two other ways:
  - `expr1 // f` (postfix notation) and `f @ expr1` (prefix notation)
- The operator `/@` is the `Map` command.
  `f /@ {1,2,5}` produces `{f[1],f[2],f[5]}`
  It maps the symbol `f` on each element of the following expression.

Matrix Construction Element-by-element II.

- Construct the following matrices:
  - Matrix/list values are defined inside curly braces `{}`,
  - colon ":" separates individual rows of a matrix.

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- User input:

```
matA = ...
matB = ...
```

Exercise

Matrix Construction Element-by-element II.

- Construct the following matrices:
  - Matrix/list values are defined inside curly braces `{}`,
  - colon ":" separates individual rows of a matrix.

$$\mathbf{A} = \begin{bmatrix} -1 & 1 \\ 1 & -2 \end{bmatrix} \qquad \mathbf{B} = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

- User input:

```
matA = ...
matB = ...
```

- ... and how does it see MATHEMATICA kernel:

```
matA//FullForm
FullForm[matB]
```

→ `List[List[-1,1],List[1,-2]]`

→ `List[List[1,2,3],List[4,5,6],List[7,8,9]]`

Exercise

Matrix Construction

- Semicolon placed at the end of a command suppresses the display of the output in the output cell.

```
a = 1
b = 5;
```

- When there is more than one command on the same line, a semicolon separates them.

```
a = 1; b = 5;
```

- To clear a symbol/variable, use `Clear` or `ClearAll` functions/commands. (`ClearAll` removes not only values but also attributes of the symbol)

`Clear[a]` or `a=.`

- MATHEMATICA is not as strict as MATLAB in defining vectors as row or column vectors. *E.g.* the following matrix multiplication works:

```
mat={{1,-1},{2,2}};
v1={4,3}; v2={{4},{3}};
MatrixForm/@{ v1.mat, mat.v1, mat.v2}
```

. . . but the following does not work

```
v2 . mat;
```

Basic Math Operators I.

- Operator types:
  - arithmetic:
    - matrix,
    - vector,
  - relational,
  - logical and other (to be mentioned later ...).
- Other operations using MATHEMATICA functions:
  - `Conjugate`, `ConjugateTransposе`
  - `Sum`, `Total`, `Det` – determinant, `Sqrt` – square root,
  - and hundreds of other functions ...

Notice the operator's precedence (to be discussed later).

`+`, `Plus`	addition
`-`, `Minus`	subtraction
`*`, `Times`	multiplication
`^`, `Power`	power
`Transpose`	transpose
`LinearSolve`	left matrix division

Basic Math Operators II.

Type in the following commands:

- ```
  a3 = -2/4;
  a4 = -0.5;
  a5 = -.5000000000000000000000000000;
  ```
- What is the difference between a_3, a_4 and a_5? Evaluate function **Accuracy** for each variable/symbol.

  ...

  ...

  ...

Exercise
  ```

Basic Math Operators II.

Type in the following commands:

- ```
  a3 = -2/4;
  a4 = -0.5;
  a5 = -.5000000000000000000000000000;
  ```
- What is the difference between a_3, a_4 and a_5? Evaluate function `Accuracy` for each variable/symbol.

  ...
  ...
  ...

  → `Infinity`
  → `16.2556`
  → `29.`
- You can put it to one list to display it on one line:

  ```
  {Accuracy[a3], Accuracy[a4], Accuracy[a5]}
  ```

  → `{Infinity,16.2556,29.}`

Exercise
  ```

Basic Math Operators II.

- ... the previous expression can be written as:
  `Accuracy /@ {a3,a4,a5}`
- Define list `myList={a3,a4,a5};` and evaluate functions `Precision`, `Head`, `FullForm`, on each element of the list `myList`.
  ...
  ...
  ...
- Beware the precedence of operators: Explain the difference between `a2/a3*a4` and `a2/(a3/a4)`.
- *MATHEMATICA tries to maintain the precision of calculation as high as possible.*
  `{Sqrt[2], Sqrt[2]//N, Sqrt[2.], Tan[-1], N[Tan[-1],9], 30!, N[30!]}`
  $\rightarrow$ {$\sqrt{2}$, 1.41421, 1.41421, -Tan[1], -1.55740772,
  265252859812191058636308480000000, 2.65253*10^32}

Exercise

Basic Math Functions I.

There are groups of functions in MATHEMATICA (similar to MATLAB): Scalar functions (*e.g.*: `Sin`, `Sqrt`, `Log`, `Factorial`), Vector functions (*e.g.*: `Total`, `Max`) and Matrix functions (*e.g.*: `Det`, `Trace`).

- Every symbol (variable/function) has its values/definitions and attributes.
  - Get help (values and definitions) on a specific symbol: `?symbName`, resp. more detailed help: `??symbName`:

```
x = 5 - 4; f2[x_] := x^2;
?x
?f2
?Sin
```

  - Get attributes of a specific symbol: `Attributes`[*symbName*]:

```
Attributes[x]
Attributes[f2]
Attributes[Sin]
```

Basic Math Functions I.

- Symbol attribute **Listable** says MATHEMATICA to *automatically* distribute/map the symbol over *all elements* of the input list (recursively, at all levels) – it makes a symbol work as a scalar function (e.g. **Sin**).

```
v = {2, {2.3, -1/2}, 5, {{4 Pi, 10}, 3/8}};
h3[v]
h3 /@ v
SetAttributes[h3, Listable];
h3[v]
ClearAttributes[h3, Listable];
Sin[v]
```

- All "*scalar*" functions have attribute **Listable**.

```
Attributes /@ {Power, Sin, Cos, Plus, Times, Total, Max, Det, Trace}
```

- Example: the difference between matrix power and element-wise power.

```
matA = {{2, 3}, {1, -1}};
MatrixForm /@ {matA, matA . matA, MatrixPower[matA,2], matA^2, Power[matA,2]}
```

Basic Math Functions II.

2 Solved examples

- Using MATHEMATICA help, calculate the following expression: $a\sin^2(\alpha) + a\cos^2(\alpha) - a$
  - Use numerical values of your own choice. Additionally, functions **N** and **Simplify** may help too.

```
a=2; al=1; a Sin[al]^2 + a Cos[al]^2 - a //N
Clear[a,al]; a Sin[al]^2 + a Cos[al]^2 - a //Simplify
```

- Evaluate the following expression: $\log_{10}(a) + \log_{10}(b) - \log_{10}(ab)$

```
a = 3.1; b = 0.7; Log10[a] + Log10[b] - Log10[a b]
Clear[a, b]; Assuming[a>0 && b>0, Log10[a] + Log10[b] - Log10[a b] // Simplify]
```

- 1 Unsolved example. Find the sum of all elements in individual rows of the following matrix:

$$T = \begin{bmatrix} \frac{1}{2} & \frac{1}{3} & \frac{1}{4} & \frac{1}{5} \\ 6 & 7 & 8 & 9 \\ 0.2 & 0.3 & 0.4 & 0.5 \end{bmatrix}.$$

```
matT = {{1/2,1/3,1/4,1/5},{6,7,8,9},{0.2,0.3,0.4,0.5}};
...
```

Exercise

Basic Math Functions III.

- Assume following vectors $\mathbf{u} = (1, 2, 3)$ and $\mathbf{v} = (3, 2, 1)$.
  - Calculate:

    scalar $\mathbf{u}\mathbf{v}^{\mathrm{T}}$ $\mathbf{v}\mathbf{u}^{\mathrm{T}}$ scalar

    matrix $\mathbf{v}^{\mathrm{T}}\mathbf{u}$ $\mathbf{u}^{\mathrm{T}}\mathbf{v}$ matrix
  - Following functions are needed:
    - `.` (resp. `Dot`)– matrix/vector multiplication,
    - `Outer` – outer product.
    - Note: `Transpose` will not help.

Exercise

```
u = {1, 2, 3}; v = {3, 2, 1};
```

...
...
...

$$10 \qquad\qquad 10$$

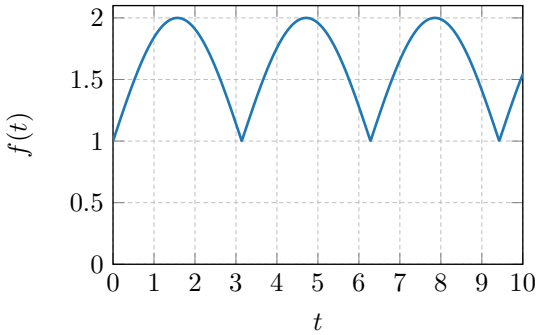
$$\begin{bmatrix} 3 & 6 & 9 \\ 2 & 4 & 6 \\ 1 & 2 & 3 \end{bmatrix} \qquad \begin{bmatrix} 3 & 2 & 1 \\ 6 & 4 & 2 \\ 9 & 6 & 3 \end{bmatrix}$$

Plot a Function

- Plot the following function in the interval $t \in (0, 10)$.
  - Use **`Plot`** command.
  - Do not define function $f(t)$ separately. Write the formula directly to the **`Plot`** command.

$$f(t) = |\sin(t)| + 1$$

...
or
...
...
...
...
...



Exercise

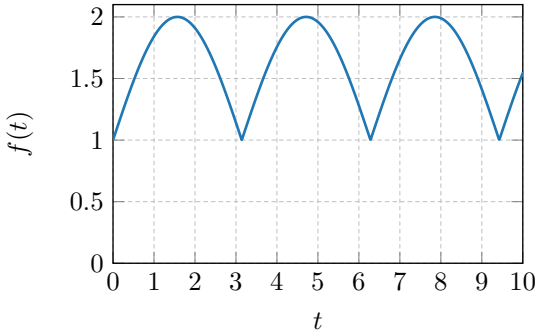
Plot a Function

- Plot the following function in the interval $t \in (0, 10)$.
  - Use **Plot** command.
  - Do not define function $f(t)$ separately. Write the formula directly to the **Plot** command.

$$f(t) = |\sin(t)| + 1$$

... and the MATLAB-like approach in MATHEMATICA

```
tVec = Subdivide[0, 10, 20];
fVals = Abs[Sin[tVec]]+1;
ListLinePlot[Transpose[{tVec, fVals}],
  PlotRange ->{{0,10},{0,2.1}},
  GridLines ->{Range[0,10],
    Table[0.5 i,{i,0,5}]},
  FrameLabel -> {"t", "f(t)"}, Frame ->
    True]
```



Exercise

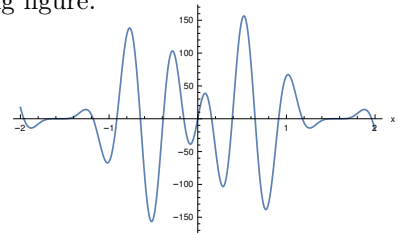
Questions?

A8B17CAS
lukacjo1@fel.cvut.cz

November 25
Winter semester 2024/25

Voluntary homework

1. Compute fifth derivative of the expression/function $\sin^6(x)\cos^{10}(x)$, assign the results to a symbol **`expr`**, and plot the expression **`expr`** from -2 to 2. (hint: use **`D`** command))
   You should get the following figure.



2. Define a matrix and compute the minimum of each row and each column. Use the following matrix.

$$\begin{bmatrix} 4 & -2 & 0 & 8 \\ 3 & -1 & 2 & 5 \\ -11 & 13 & 6 & 7 \end{bmatrix}$$

   Hint: recall that a matrix is a list of rows: **`{row1,row2,row3}`**. Use **`Min`** command.

Voluntary homework

3. Compute limit of $(1+nx)^{1/x}, n \in \mathbb{N}$ for $x \to 0$. The expressions are stored in a list/vector **vecA**. You can use the **Limit** command.

```
Clear[x]; nMax = 10;
vecA = Table[(1 + n x)^(1/x), {n, nMax}]
```

4. For a given vector of numbers, find the minimum distance of the numbers to 3. *E.g.* for the vector `{-10, 9, -7, 2, 0}`, the result should be 1 because the closest to 3 is 2 with distance 1.

```
nNums = 5;
nMin = -10; nMax = 10;
vecB = RandomInteger[{nMin, nMax}, nNums]
```

5. Numerically compute the area of the shaded region in the figure. Use **NIntegrate**.