Symbolische Programmierung

Wolfram Mathematica, sympy, ...

Ruben 14felgenh

10. Januar 2023

Disclaimer

- Ich bin kein Mathematica-Profi
- Ich bin kein sympy-Profi
- Ich laber möglicherweise Müll

Installation

Mathematica:

- 72 Lizenzen für Mathematica 13.1 verfügbar
- Verbindung zum Lizenzserver erforderlich
- Mehr Infos (nur aus dem Uni-Netz erreichbar):
  `https://www.rrz.uni-hamburg.de/services/software/software-thematisch/software-alpha/mathematica.html`

SymPy:

- `pip install sympy`
- Für die volle Fake Mathematica Experience:
  - `pip install jupyterlab`

Symbolische vs. Numerische Programmierung

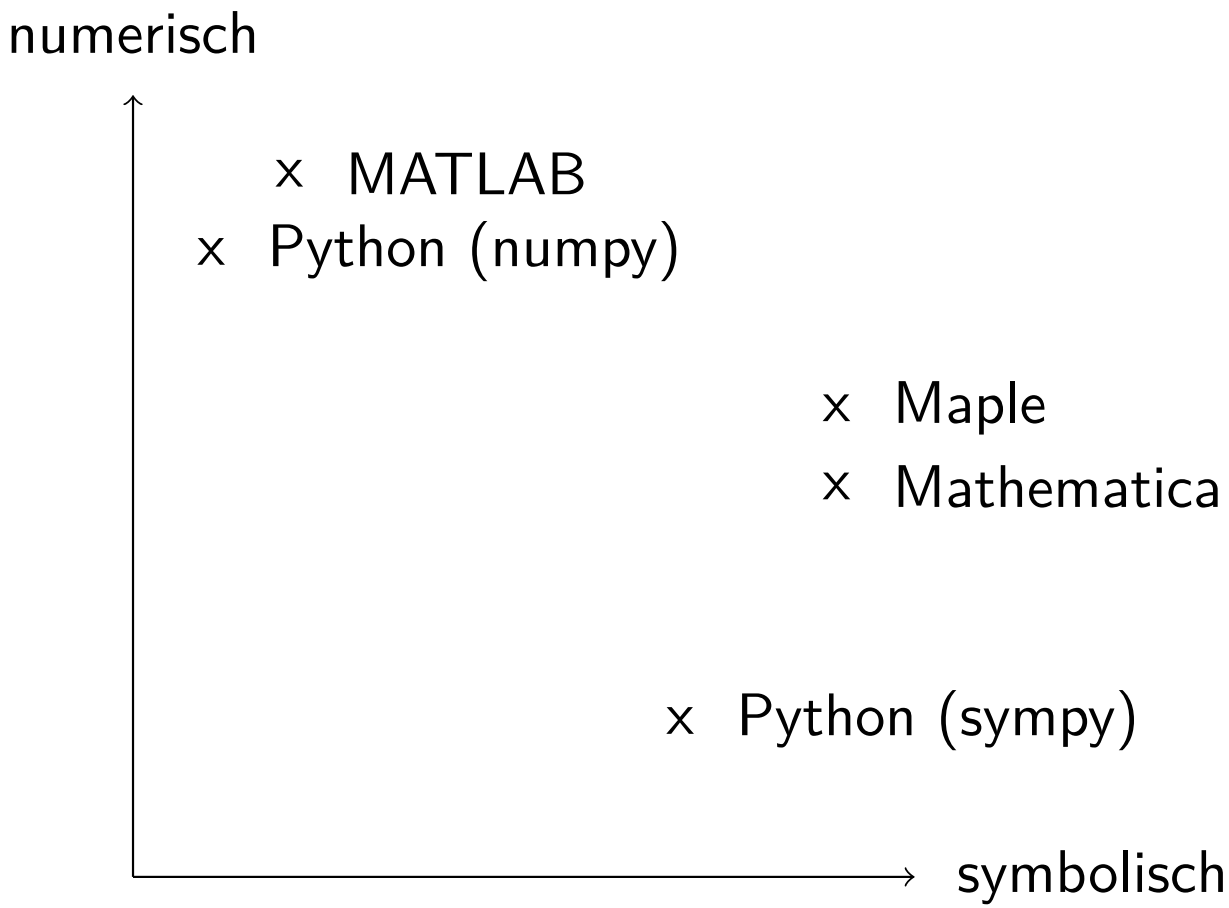
Wir wollen:

- Gleichungen umstellen, lösen, vereinfachen
- Ableitungen, Stammfunktionen bestimmen
- Eigenwerte, Gleichungssysteme
- ...

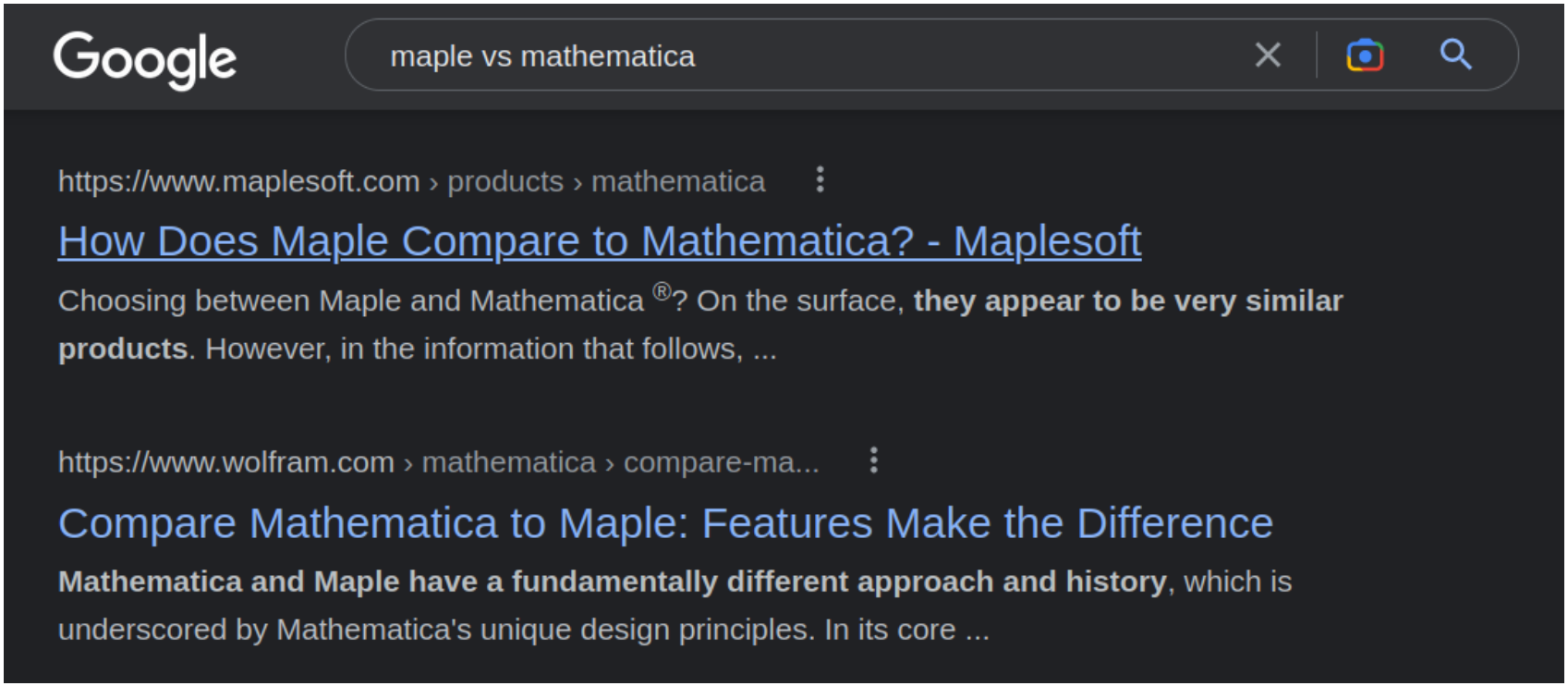
Analytische Mathematik $\leftrightarrow$ Numerische Mathematik

Symbolische Programmierung $\leftrightarrow$ Numerische Programmierung

Programmiersprachen



Programmiersprachen



Programmiersprachen

Mathematica

− Kostet Geld

− Neue Programmiersprache

\+ Elegantere Syntax

\+ Mehr und ausgereiftere Features

sympy

\+ Gratis

\+ Vertraute Syntax

− Etwas clunky

− Etwas basic

Beispiele (Python)

```
>>> import numpy as np

>>> np.sqrt(2)
1.4142135623730951



>>> np.sqrt(2)**2
2.0000000000000004
```

```
>>> import sympy as sp

>>> sp.sqrt(2)
sqrt(2)

>>> sp.N(sp.sqrt(2))
1.41421356237310

>>> sp.sqrt(2)**2
2
```

Beispiele (Python)

:(

```
>>> import sympy as sp

>>> a,b = sp.symbols("a b")

>>> 5*a + 4*b - 2*a
3*a + 4*b

>>> sp.expand((a + b)**2)
a**2 + 2*a*b + b**2
```

Beispiele (Python)

Berechne $\int_0^1 x + x^2 \, \mathrm{d}x$.

```
>>> import numpy as np

>>> def f(x):
...     return x + x**2

>>> xs = np.linspace(0, 1,
    ↪ num=100)



>>> np.trapz(f(xs), x=xs)
0.833350338400843 4
```

```
>>> import sympy as sp

>>> x = sp.symbols("x")

>>> f = x + x**2

>>> sp.integrate(f)
x**3/3 + x**2/2

>>> sp.integrate(f, (x,0,1))
5/6

>>> 5/6
0.833333333333334
```

Beispiele (Python)

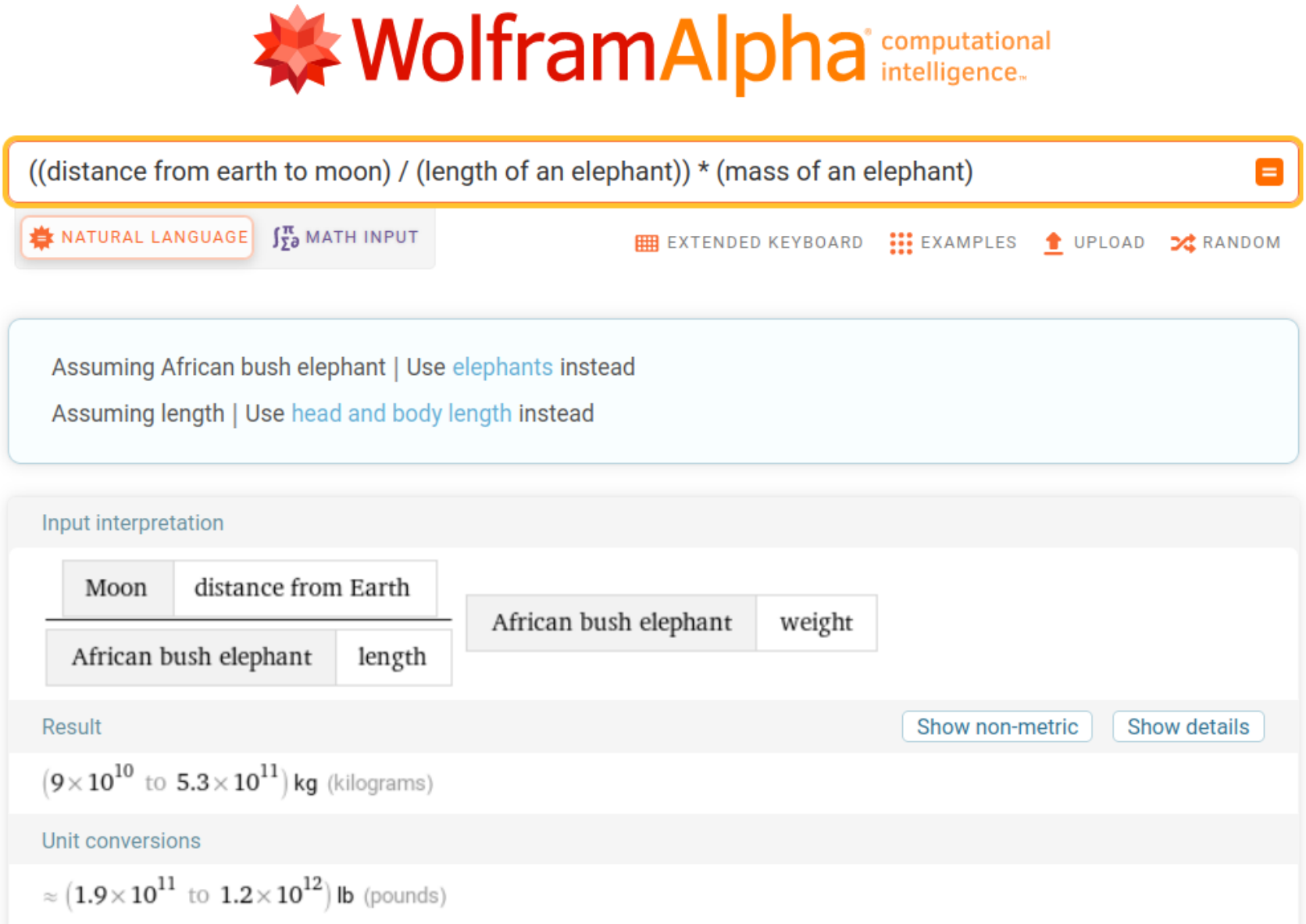
Berechne $\int_5^{10} \frac{x}{\sqrt{x^4 + 10x^2 - 96x - 72}} \, \mathrm{d}x.$

```
>>> import numpy as np

>>> def f(x):
...     return x /
    ↪ np.sqrt(x**4 +
    ↪ 10*x**2 - 96*x - 72)

>>> xs = np.linspace(5,
    ↪ 10, num=100)

>>> np.trapz(f(xs), x=xs)
0.7579090249485106
```

```
>>> import sympy as sp
>>> x = sp.symbols("x")

f = x / sp.sqrt(x**4 +
    ↪ 10*x**2 - 96*x - 72)

>>> sp.integrate(f)
Integral(x/sqrt(x**4 +
    ↪ 10*x**2 - 96*x - 72),
    ↪ x)

>>> sp.N(sp.integrate(f,
    ↪ (x,5,10)))
0.75787697605757 5
```

https://en.wikipedia.org/wiki/Risch_algorithm

Wolfram|Alpha



Wolfram|Alpha

In[4]:= `WolframAlpha["((distance from earth to moon) / (length of an elephant)) * (mass of an elephant)"]`

Out[4]=

Assuming African bush elephant | Use elephants instead
Assuming length | Use head and body length instead

Input interpretation:

$$\frac{\text{Moon}\ \ \text{distance from Earth}}{\text{African bush elephant}\ \ \text{length}} \quad \text{African bush elephant}\ \ \text{weight}$$

Result: Show non–metric | Show details

$(9\times10^{10}$ to $5.3\times10^{11})$ kg (kilograms)

Unit conversions:

$\approx (1.9\times10^{11}$ to $1.2\times10^{12})$ lb (pounds)

$\approx (9\times10^{13}$ to $5.5\times10^{14})$ grams

$\approx (9\times10^{7}$ to $5.5\times10^{8})$ t (metric tons)

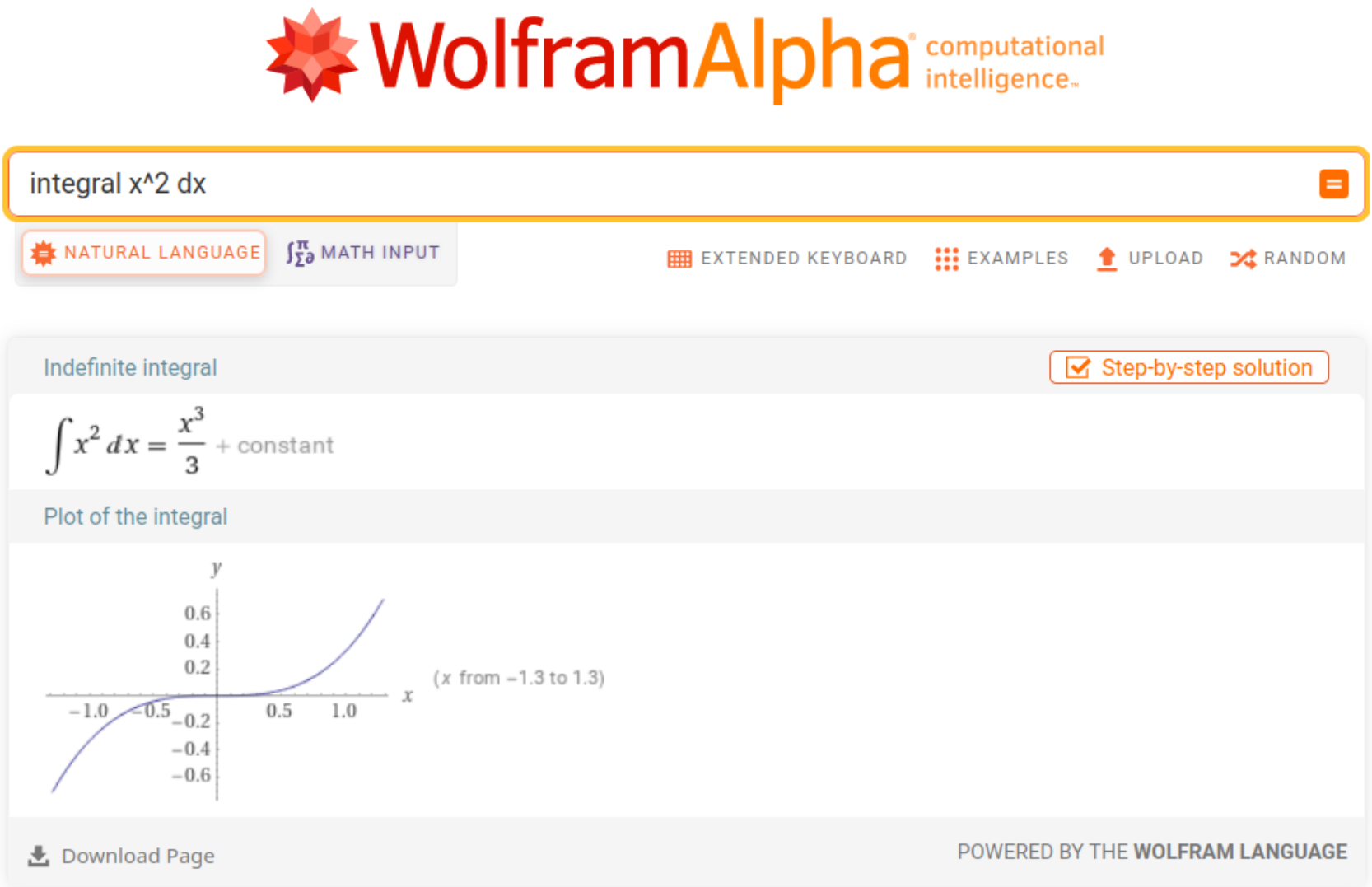
WolframAlpha

Wolfram|Alpha

In[12]:=
```
Integrate[x^2, x]
```

Out[12]=
$$\frac{x^3}{3}$$

Wolfram|Alpha



Wolfram|Alpha

WolframAlpha computational intelligence

$\int x^2 \, d\, x$

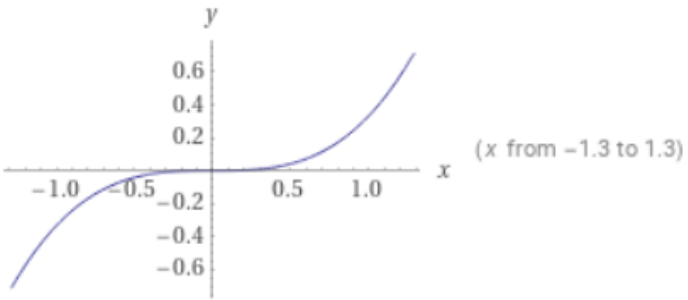
NATURAL LANGUAGE | MATH INPUT

Indefinite integral

Step-by-step solution

$$\int x^2 \, dx = \frac{x^3}{3} + \text{constant}$$

Plot of the integral



(x from −1.3 to 1.3)

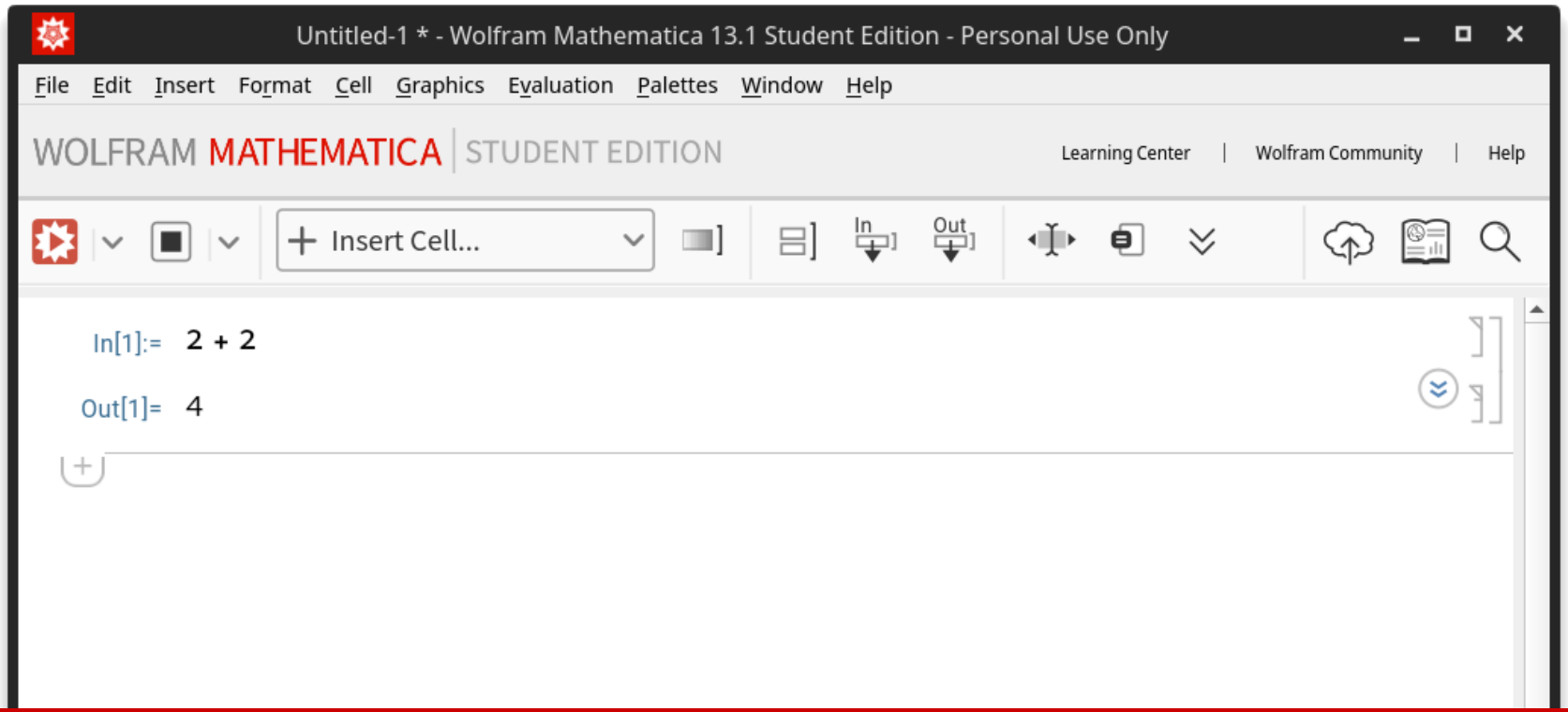
POWERED BY THE **WOLFRAM LANGUAGE**

Mathematica

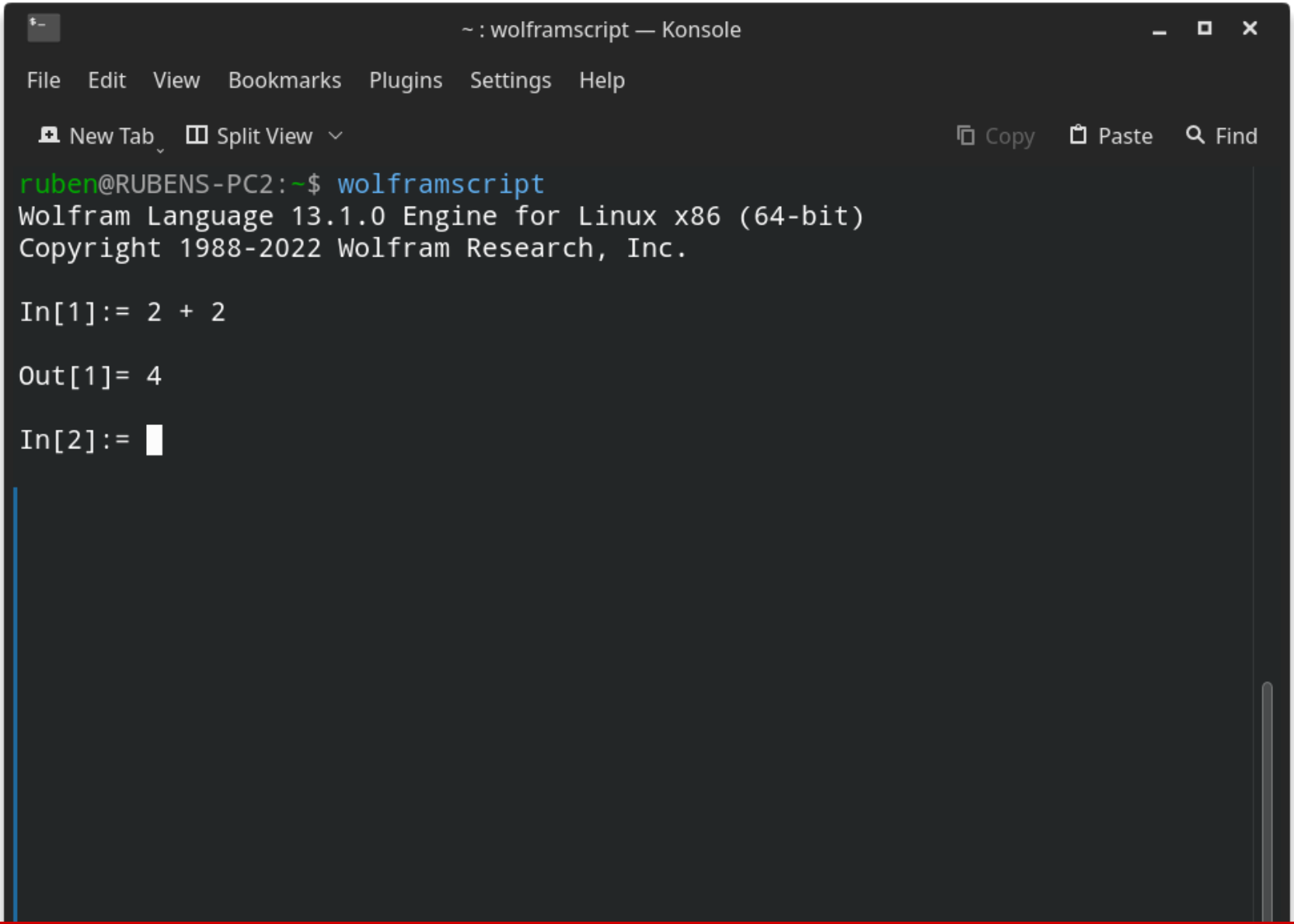
- Interface für die Wolfram Language
- Kernel und GUI (vgl. Jupyter Notebook)
- Interaktion über Notebook
- Ein- und Ausgaben sind strukturiert in Cells

Eine erste Rechnung

- Im Hauptbildschirm: *New Document* → *Notebook*
- Tippe „2+2“ und bestätige mit Shift + Enter



Alternativ: Über die Konsole



Intermission

sympy

- `https://www.sympy.org/en/features.html`
- `https://jupyter.org/`
- `jupyter-lab`

Übung

Betrachte die Funktionenfolge

$$f_n(x) = \cos\left(\frac{\pi\, x^{(2n+1)}}{2^{(3n-7)}}\right).$$

Finde $n \in \{1, \ldots, 25\} \subset \mathbb{N}$ so, dass die betragsmäßig kleinste Nullstelle von $f_n(x)$ bei $x = 2$ liegt
(*also* $f_n(2) = 0$ *und* $f_n(x) \neq 0$ *für alle* $x \in (-2, 2) \subset \mathbb{R}$).