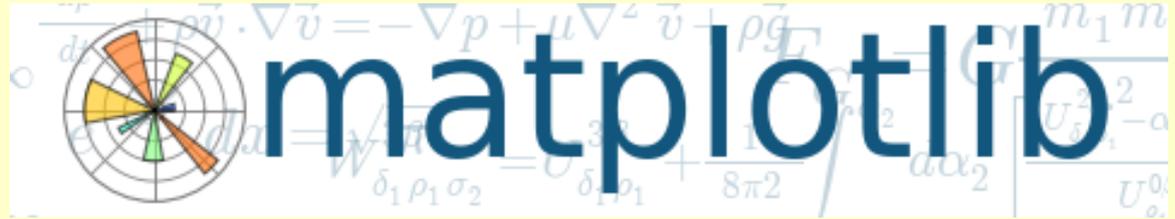


NumPy



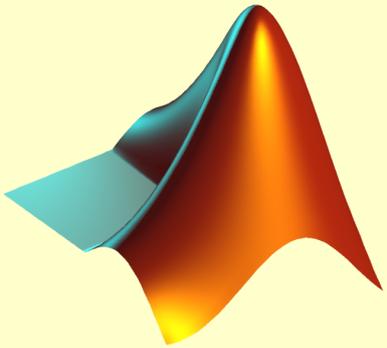
Wissenschaftliche Grafiken und Berechnungen mit Python



24. + 25. April 2015 www.linuxtage.at

Dietmar Thaler, 8911 Admont
dietmar.thaler@posteo.at
Version 24.04.2015

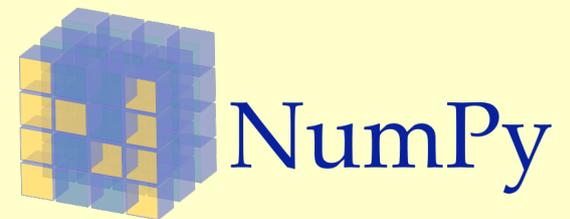
Motivation



MATLAB (TM) von *The MathWorks, Inc.* :
das ultimative Tool für
„**number crunching**“ und
wissenschaftliche Grafik

Aber: **Standard-Lizenz MATLAB**
ca. EUR 2000,- (Home- und
Studentenlizenz inzwischen **viel**
günstiger)

NumPy und Matplotlib unter Python
sind eine **freie Alternative zu MATLAB**



Gliederung

0. Warum Python? - Warum Python 2.7?

1. Was sind Matplotlib und Numpy?

2. Ipython als interaktive Shell mit pylab als matplotlib-numpy-Wrapper

3. Pyplot und Numpy in einem kleinen Programm

4. Komplexere(s) Beispiel(e)

5. Erweiterungen und Toolkits: z.B. „Scipy“ und „Basemap“

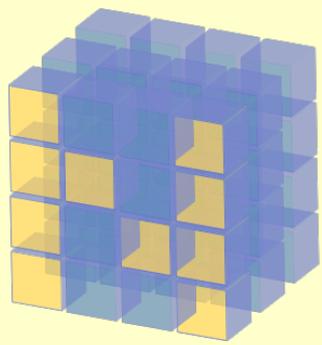
Warum Python?



- Leicht zu lernen und flexibel
- Hinreichend schnell
- Freie Software
- Viele Erweiterungen

Warum Python 2.7 (und nicht 3.x)?

- Alte Python Software läuft unverändert
- Python 2.7 ändert sich nicht mehr
- ~~Viele~~ Manche Programmbibliotheken sind noch nicht 3.x kompatibel



NumPy

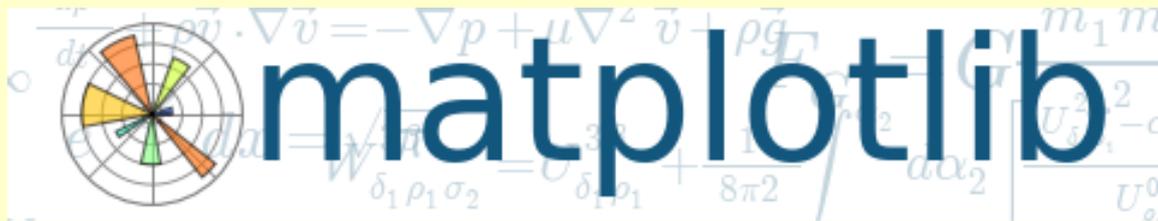
<http://www.numpy.org/>

NumPy ist eine grundlegende Programmierbibliothek für **wissenschaftliches Rechnen unter Python**.

- Leistungsstarke **N-dimensionale Arrays** (Felder, Matrizen)
- dazu vielseitige Funktionen zur **Manipulation von Arrays**
- Nützliche **Funktionen**: Lineare Algebra, Fourier Transformation, Zufallszahlen, ... (neben den Erweiterungen der Standard-Mathematik-Bibliothek auf Arrays)

NumPys Flexibilität erlaubt die schlüssige **Integration in weiterführende Bibliotheken**, z.B. in **Matplotlib** sowie die Integration von C/C++ und Fortran.

BSD Lizenz (<http://www.numpy.org/license.html#license>)



<http://matplotlib.org/>

Matplotlib ist eine Programmbibliothek für **technisch-wissenschaftlich-mathematische Grafiken** aller Art.

- begründet von John D. Hunter *1968 - +2012
- Objektorientiert und prozedural verwendbar
- 2D- und 3D-Funktionalität (z.T. durch Toolboxes)
- basierend auf Numpy
- diverse grafische Backends zur Darstellung verfügbar



BSD kompatibler Code in der Python Software Foundation Lizenz (PSF, <https://www.python.org/psf/>)

3 Modi von matplotlib

- **Interaktiver Modus (Pylab-Modus):**
 - Analog zu Matlab
 - Vorzugsweise in Ipython
 - Ungünstiger in der Programmierung (Verwischung der Matplotlib- und Numpy-Funktionalität)
- **Prozeduraler Modus**
 - matplotlib.pyplot und numpy
 - Funktionalität in etwa wie im Pylab-Modus
 - saubere Programmierbarkeit (getrennter name space)
- **Objektorientierter Modus:**
 - Volle Kontrolle, komplex und daher oft unübersichtlich, (leider) gelegentlich notwendig

IP[y]: IPython
Interactive Computing

<http://ipython.org>

mit **pylab**
(interaktiver Matplotlib- und
NumPy-Wrapper)

- IPython ist eine komfortable Python-Shell
- Installation (unter Ubuntu 14.4):

```
>sudo apt-get install python-matplotlib ipython
```

NumPy und anderes wird über Abhängigkeiten mit installiert

- In der Konsole im interaktiven pylab-Modus starten:

```
>ipython --pylab
```

```
? -> Introduction and overview of IPython's features
%quickref -> Quick reference.
help -> Python's own help system.
object? -> Details about 'object', use 'object??' for extended help.
Using matplotlib backend: TkAgg
```

```
In [1]: x = linspace(-7,+7,101)
```

```
In [2]: y = sin(x)
```

```
In [3]: plot(x,y)
```

```
Out[3]: [<matplotlib.lines.Line2D at 0x7fe6919d34d0>]
```

```
In [4]: grid()
```

```
In [5]: xlabel('x')
```

```
Out[5]: <matplotlib.text.Text at 0x7fe6913ef110>
```

```
In [6]: ylabel('$\sin(x)$')
```

```
Out[6]: <matplotlib.text.Text at 0x7fe69140d090>
```

```
In [7]: title('PyLab-Testplot')
```

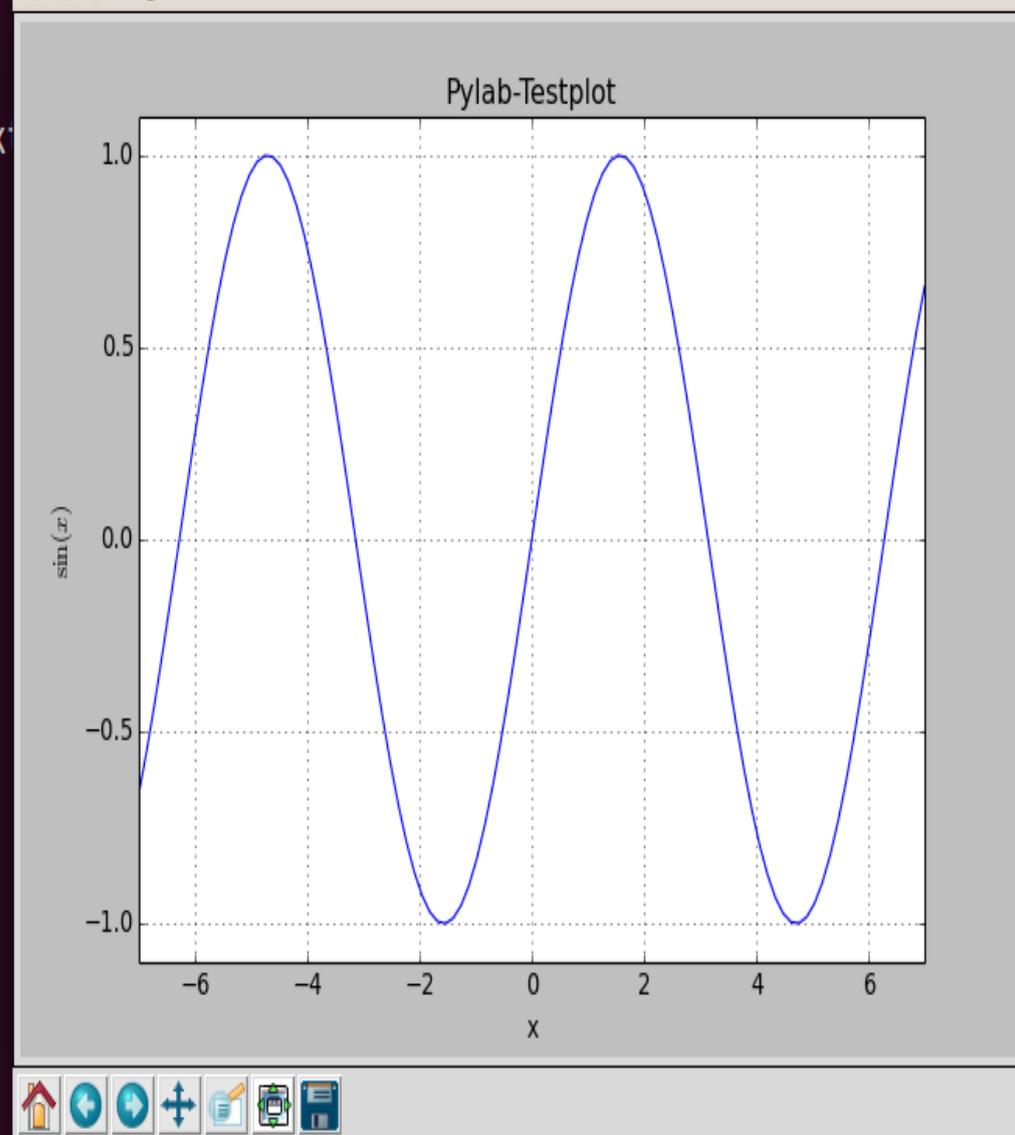
```
Out[7]: <matplotlib.text.Text at 0x7fe691a1fd10>
```

```
In [8]: ylim(-1.1,+1.1); xlim(-7,+7)
```

```
Out[8]: (-7, 7)
```

```
In [9]: █
```

Figure 1



```
In [8]: ylim(-1.1,+1.1); xlim(-7,+7)
```

```
Out[8]: (-7, 7)
```

```
In [9]: y2 = cos(x)
```

```
In [10]: plot(x,yr,'r',lw=2)
```

```
NameError                                 Traceback (most recent call last)
```

```
<ipython-input-10-475ffa8d0b37> in <module>()
```

```
----> 1 plot(x,yr,'r',lw=2)
```

```
NameError: name 'yr' is not defined
```

```
In [11]: plot(x,y1,'r',lw=2)
```

```
NameError                                 Traceback (most recent call last)
```

```
<ipython-input-11-0efd75104493> in <module>()
```

```
----> 1 plot(x,y1,'r',lw=2)
```

```
NameError: name 'y1' is not defined
```

```
In [12]: plot(x,y2,'r',lw=2)
```

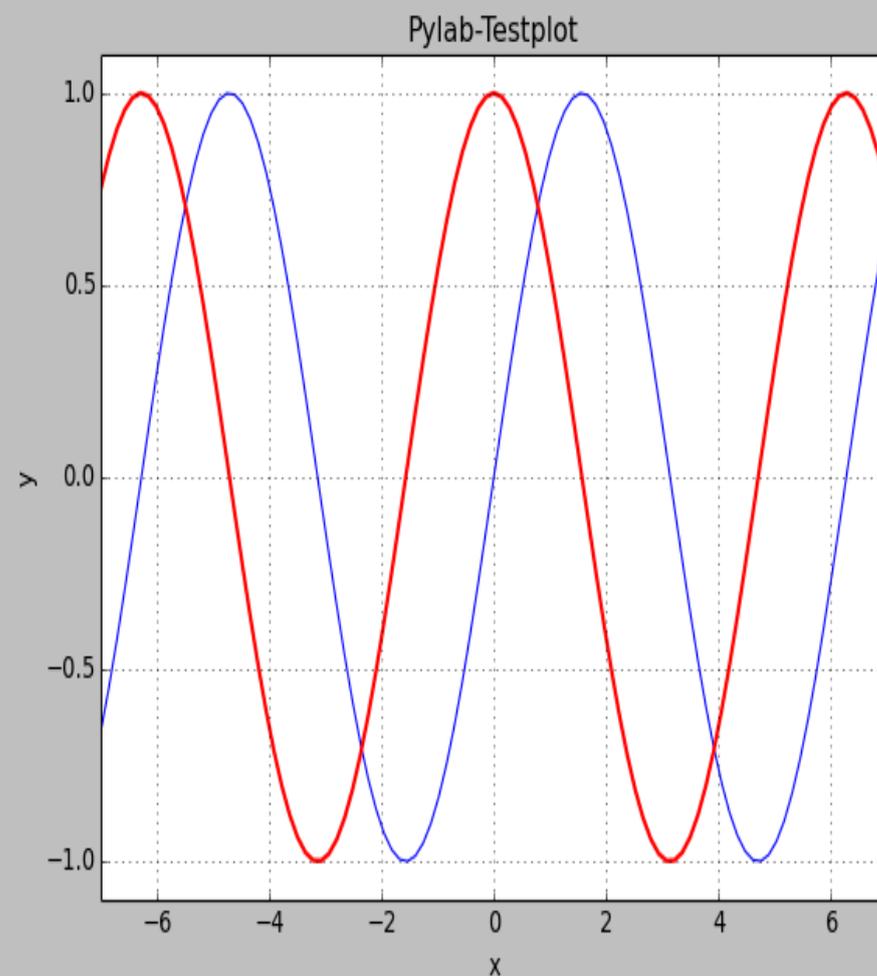
```
Out[12]: [<matplotlib.lines.Line2D at 0x7fe6919a9a10>]
```

```
In [13]: ylabel('y')
```

```
Out[13]: <matplotlib.text.Text at 0x7fe69140d090>
```

```
In [14]: █
```

Figure 1



```
In [43]: plot(x,y,'b',label='$\sin(x)$',lw=2)
```

```
Out[43]: [<matplotlib.lines.Line2D at 0x7fe690b66890>]
```

```
In [44]: plot(x,y2,'r',label=u'$\cos(x)$',lw=2)
```

```
Out[44]: [<matplotlib.lines.Line2D at 0x7fe690d64150>]
```

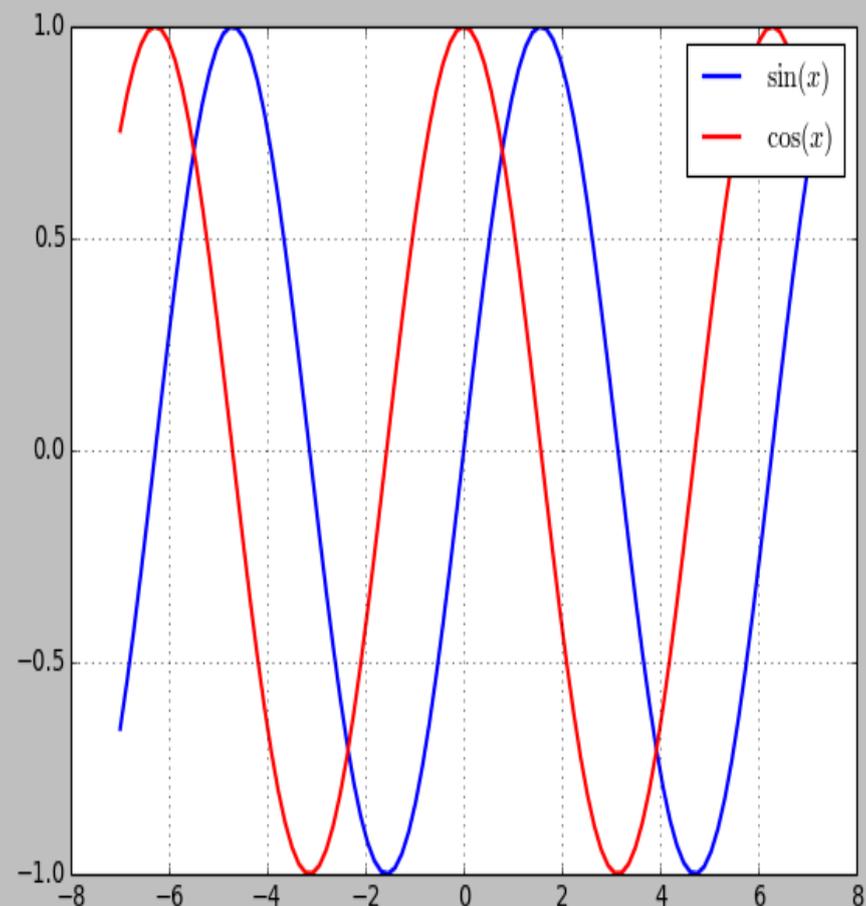
```
In [45]: grid()
```

```
In [46]: legend()
```

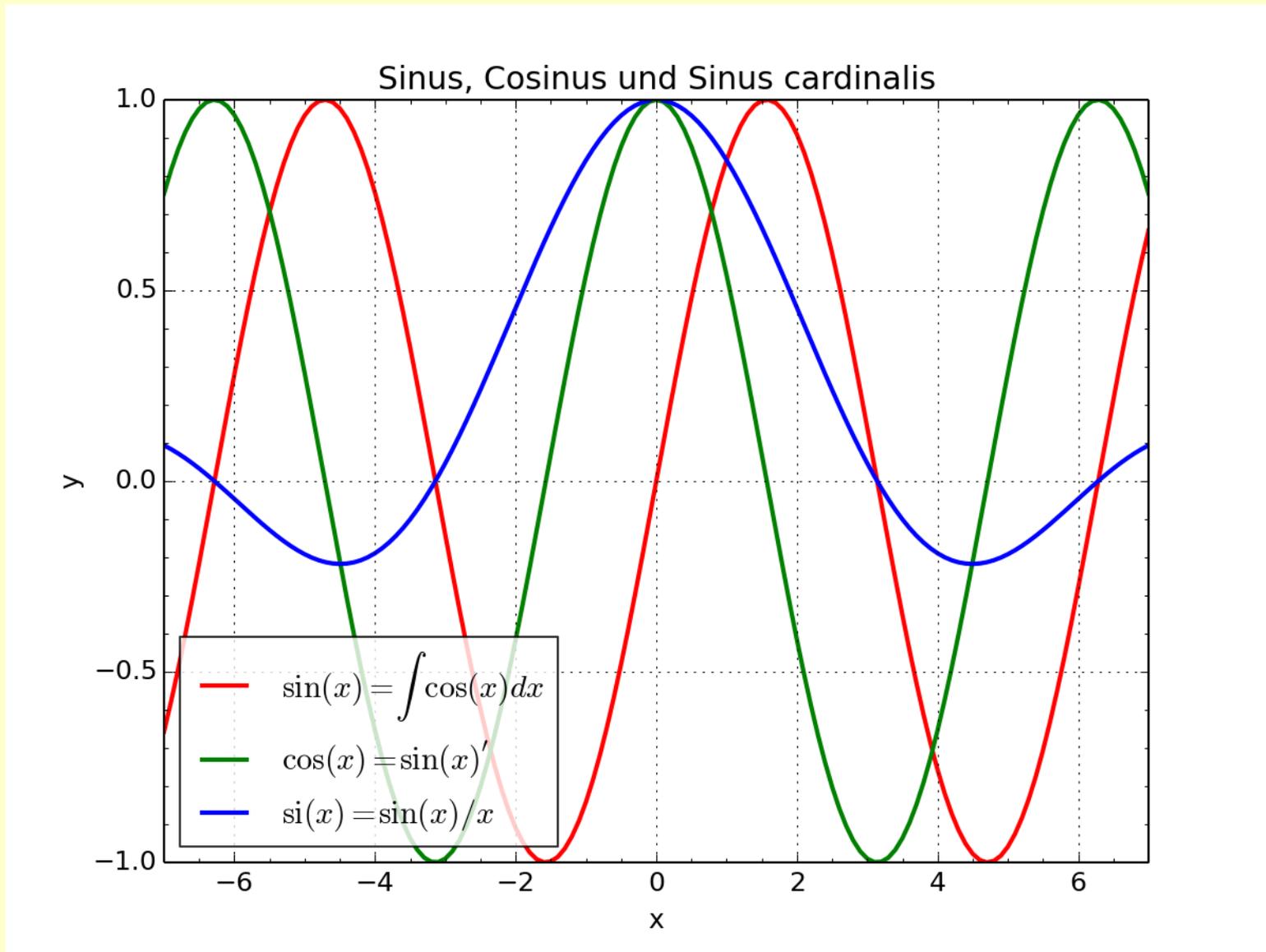
```
Out[46]: <matplotlib.legend.Legend at 0x7fe690d64890>
```

```
In [47]: █
```

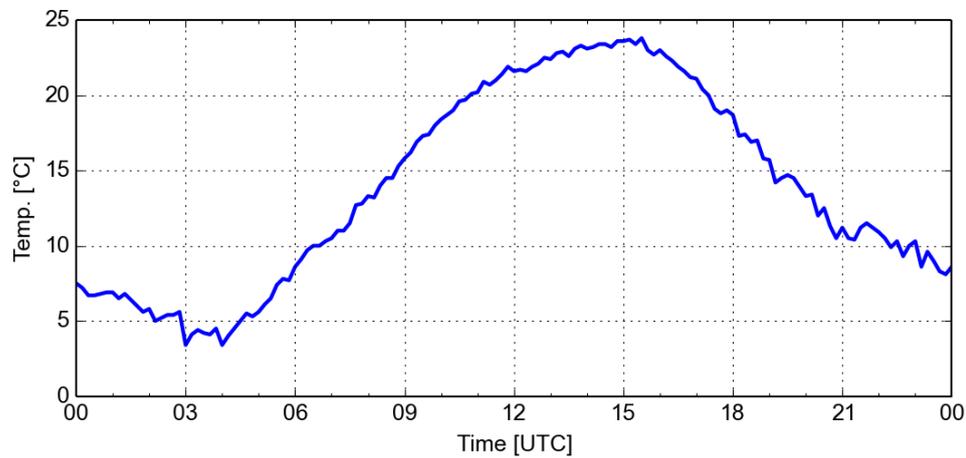
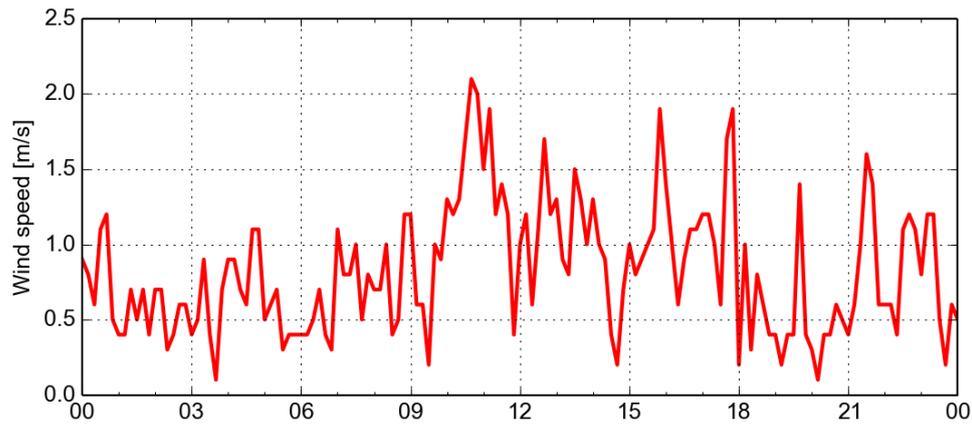
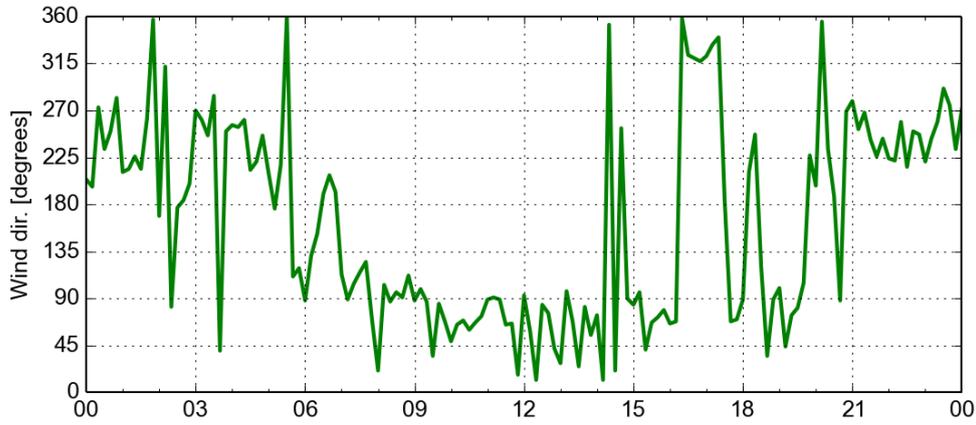
Figure 1



Ein (noch) einfaches Beispiel



STATION: 11169 DATE: 2011-04-25
(ATTENTION: anonymized test data)



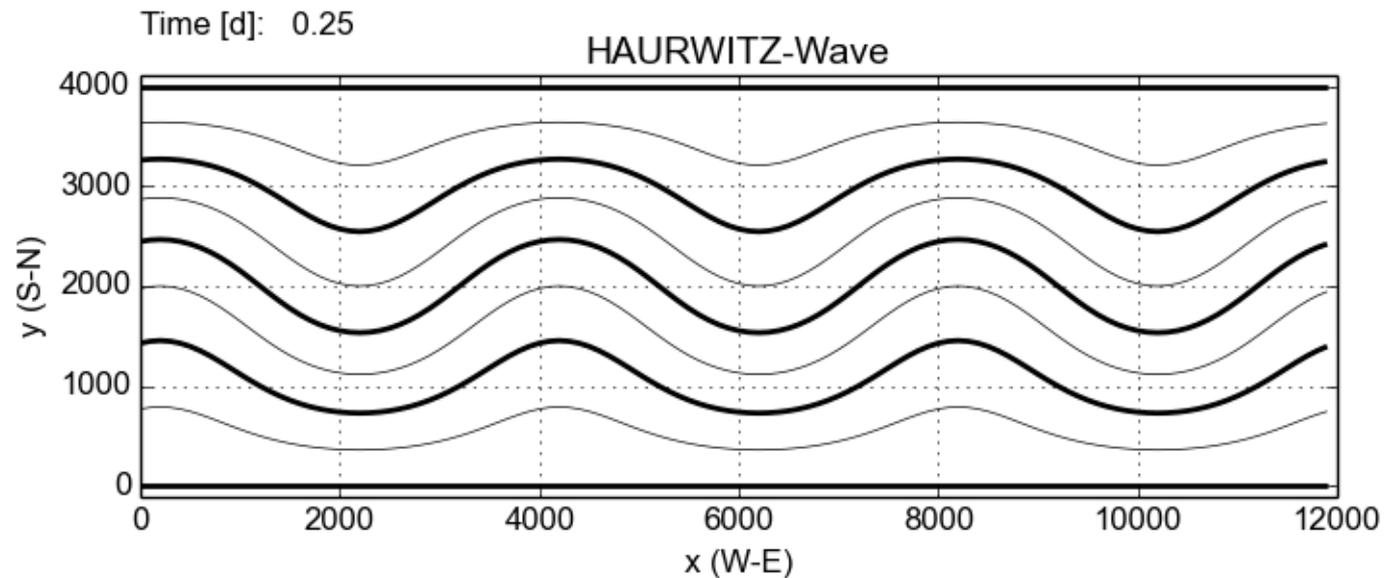
Ein komplexeres Beispiel

[plotmetdata.py](#)

Eine Animation

haurwitz_anim.avi

haurwitz.py



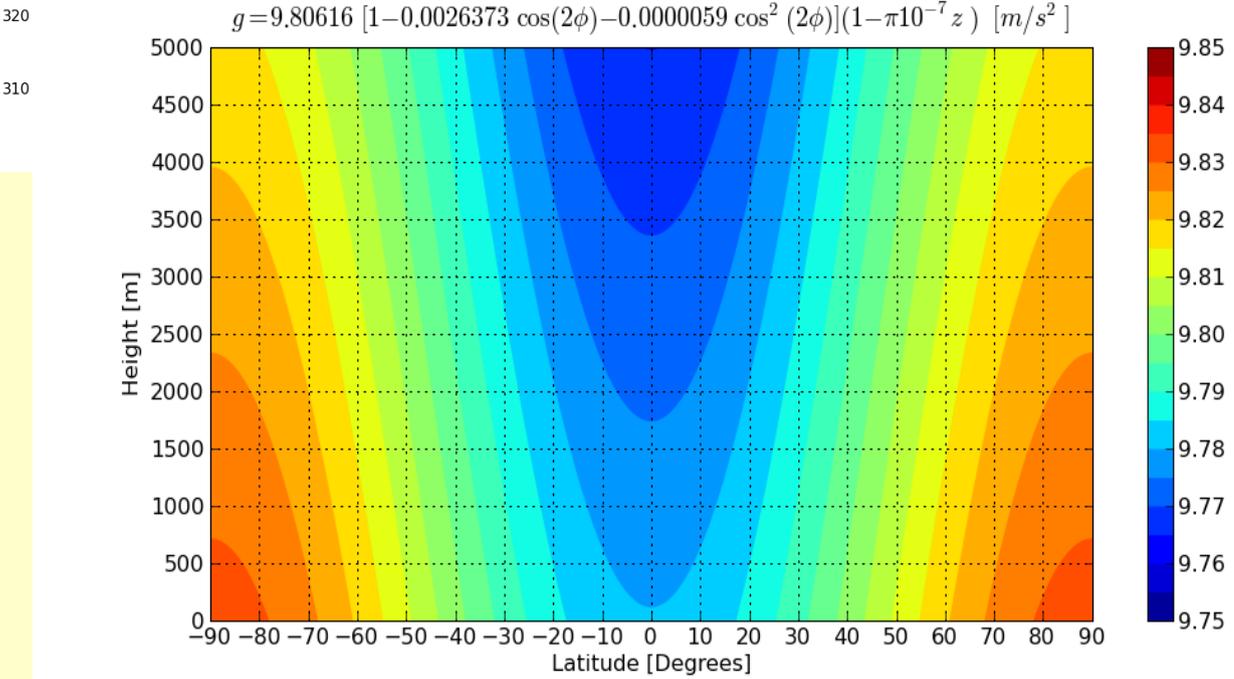
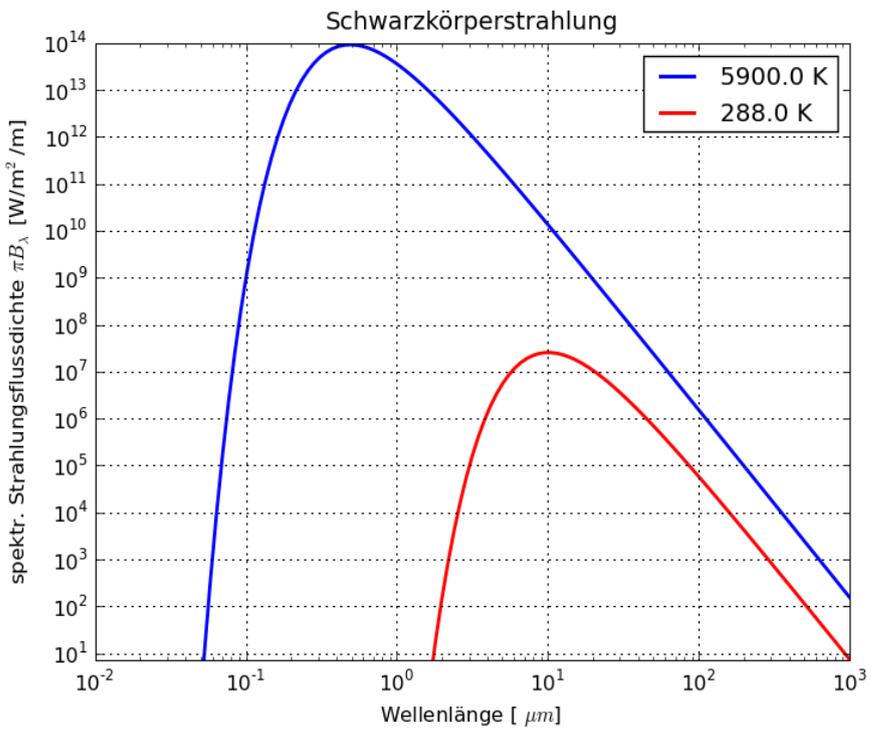
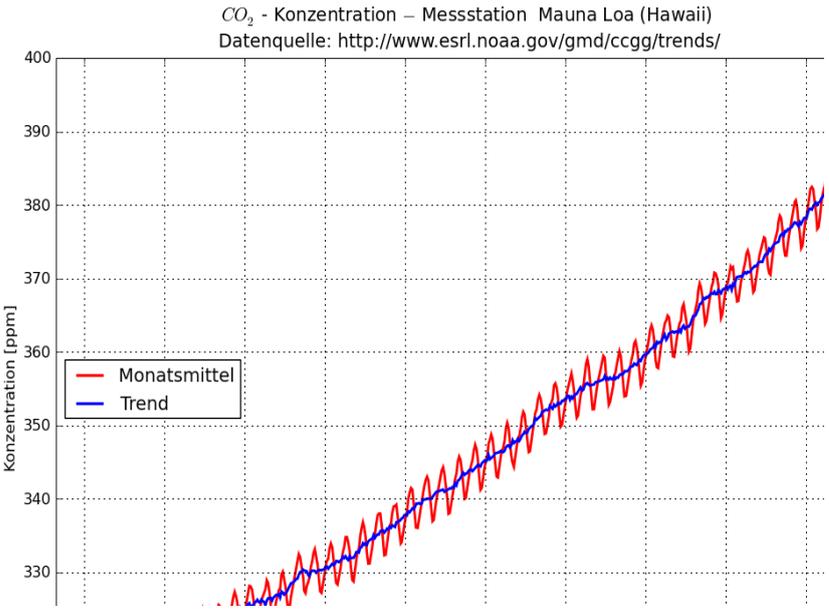
Achtung – by Loops immer:

```
plt.close() # Close
```

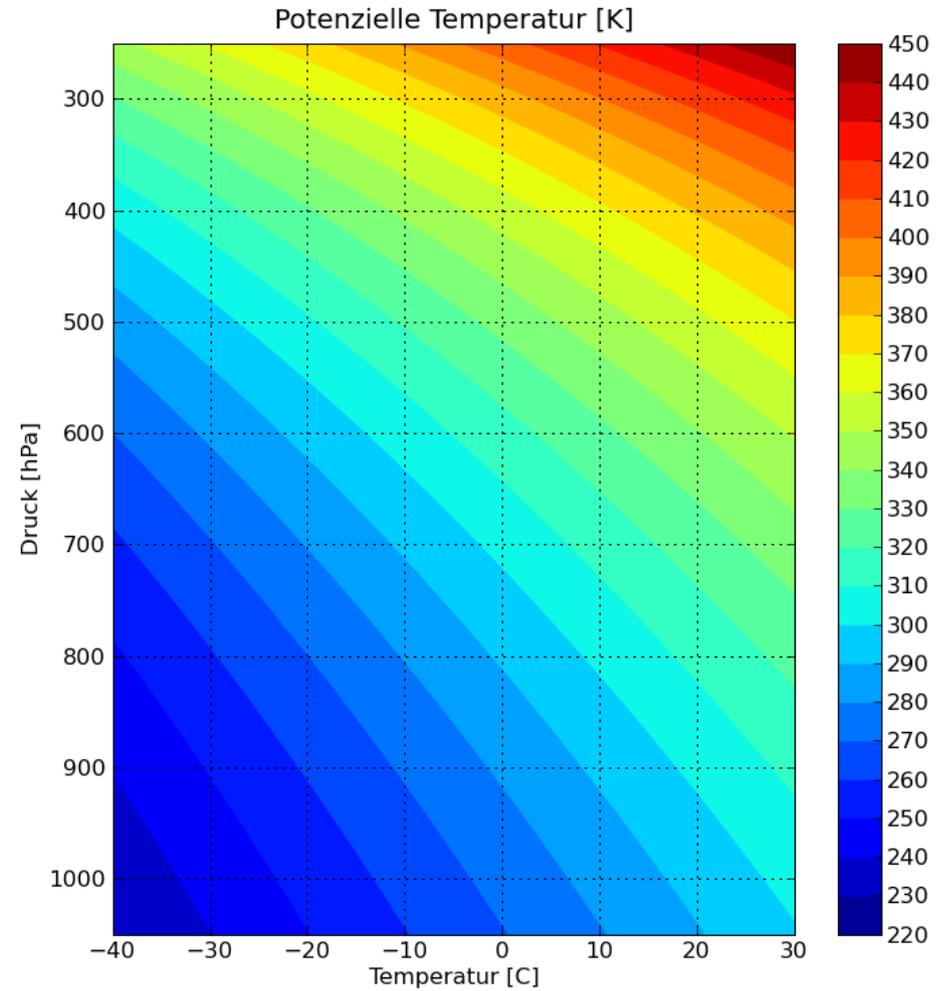
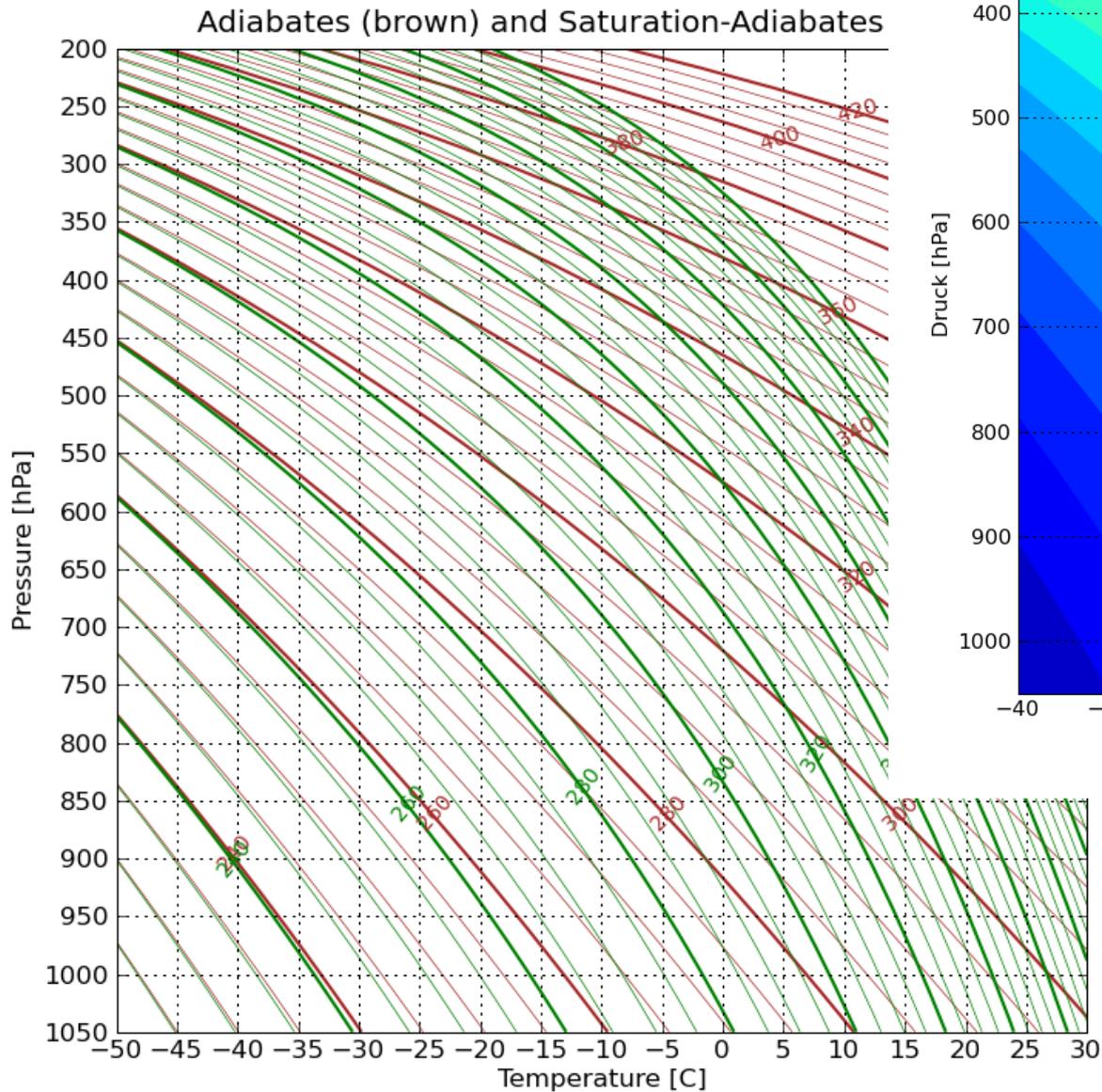
```
plt.clf() # Clear
```

Sonst ein hässliches **Speicherleck!**

Weitere Beispiele



Weitere Beispiele



.... und viele mehr in der Doku

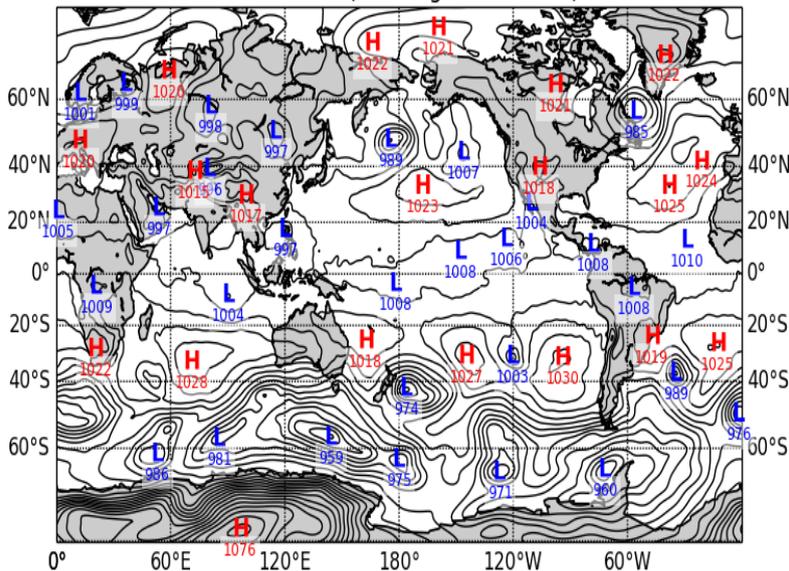
Erweiterungen und Toolkits



SciPy: <http://www.scipy.org/>

Die SciPy-Library für fortgeschrittenere Methoden der numerischen Mathematik

Mean Sea-Level Pressure (with Highs and Lows) 2013081200

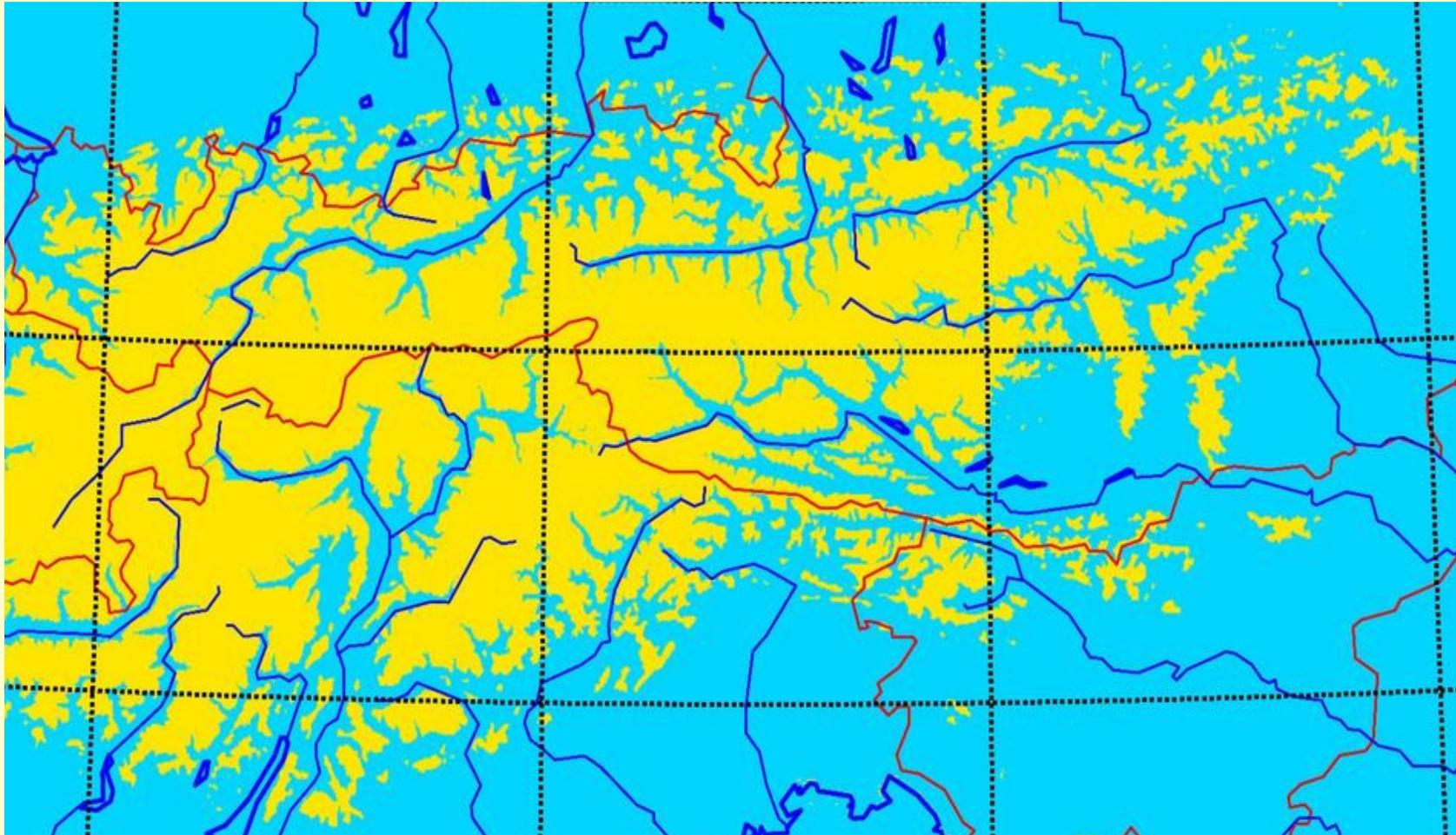


Basemap:

<http://matplotlib.org/basemap/>

Ein geographisches
Projektions- und Mapping
Toolkit

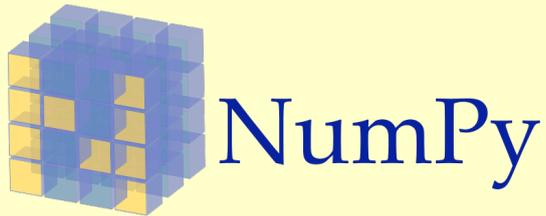
Beispiel für Basemap in Verbindung mit GIS-Routinen (http://www.osgeo.org/gdal_ogr bzw. <http://trac.osgeo.org/gdal/wiki/GdalOgrInPython>)



AustriaHD-Animation

Details auf
<http://www.foehnwall.at/alpflood.html>

Tutorials und Hilfen



Tentative NumPy Tutorial: http://wiki.scipy.org/Tentative_NumPy_Tutorial

NumPy for Matlab Users: http://wiki.scipy.org/NumPy_for_Matlab_Users

NumPy Manual: <http://docs.scipy.org/doc/numpy/>



Matplotlib Docs: <http://matplotlib.org/contents.html>

Matplotlib-users – Discussion for matplotlib users (Forum)

<https://lists.sourceforge.net/lists/listinfo/matplotlib-users>

Matplotlib und pylab – Anfängertutorial (analog zu hier)

<http://www.pro-linux.de/artikel/2/168/matplotlib-und-pylab.html>

Matplotlib in Depth:

http://www.astro.washington.edu/users/vanderplas/Astr599/notebooks/12_AdvancedMatplotlib