CgDA - Installation der Software

Release 1.2 Dec. 2020

Günter Quast

01.01.2021

Inhaltsverzeichnis

Die grafische Darstellung der Ergebnisse von experimentellen Analysen oder theoretischen Rechnungen ist in der Physik von zentraler Wichtigkeit. Dabei sind Wiederholbarkeit und Reproduzierbarkeit ganz wesentliche Anforderungen, die sich nur erfüllen lassen, wenn die grafischen Darstellungen in einer beschreibenden Sprache erstellt werden. Dazu eignet sich insbesondere auch die Skritp- und Programiersprache *python*, s. http://www.python.org, mit den Erweiterungspaketen *numpy* und *matplotlib*.

Eine Einführung bietet der Kurs „Computergestützte Datenauswertung“ (*CgDA*), der auch Methoden für die Anwendung im Physikalischen Praktikum zu klassischen Physik bereit stellt.

Das vorliegende Dokument beschreibt die Installation der wesentlichen Komponenten für die verbreiteten Platformen *Linux (Ubuntu 8.04)*, *Windows (10)* und *Mac OSX*. Bitte beachten Sie, dass manche der Zusatzpakte *python* noch in der alten Version *2.7.x* voraussetzen, die aktuelle und im Kurs verwendte Version ist *Python 3.5.x*. Außerdem wird das Textsatz-System LaTeX benötigt. Zur Anpassung von Funktionen an Daten wird das auf *python* und dem Optimierer *minuit* basierende Paket *kafe* empfohlen. Für weiterführende Anwendungen in höheren Semestern wird evtl. das am CERN entwickelte Datenanalysepaket *Root* benötigt.

KAPITEL 1

Installation unter Linux/Ubuntu

Wenn Sie die zum Kurs empfohlene Software auf einem eigenen Linux-System installieren wollen, stellen Sie sicher, dass die in der folgenden Liste spezifizierten Pakete - je nach Distribution evtl. unter einem anderen Namen - vorhanden sind.

In der virtuellen Maschine zum Kurs sind alle Pakte bereits enthalten und Sie müssen nichts nachinstallieren.

Die folgende Liste gibt die Paketenamen an, die auf einer Basis-Installation von Ubuntu zusätzlich zu installieren sind. Die allgemeine Syntax lautet:

```
sudo apt-get install <package name>
```

Paketliste:

- synaptic (package manager)
- gfortran
- g++
- emacs, joe (or other editor)
- **LaTeX packages,**
  - texlive
  - texlive-latex-extra
  - texlive-math-extra
  - texlive-science
  - texlive-lang-english
  - texlive-lang-german
  - dvipng
- kile (editor/environment for latex)
- inkscape (scalable vector graphics)
- **python 3 + some packages:**

  - python-pip
  - python-numpy
  - python-scipy
  - python-matplotlib
  - python-qt4
- gnuplot
- qtiplot
- kafe2 (Karlsruhe Fit Environment vers. 2, siehe Link https://github.com/dsavoiu/kafe2:
  - Installation von `iminuit` via `sudo pip install 'iminuit<2'`,
  - dann `pip install kafe2`

Zur Verwendung der *root*-Klasse *TMINUIT*, zunächst *root* installieren, (s. unten) und den Eintrag *minimizer-to-use* in der *kafe*-Konfigurationsdatei *kafe.conf* von *iminuit* auf *root* setzen.

- **CERN Datenanalyse-Paket *root*:**
  - root-system
  - libroot-bindings-python5.34, libroot-bindings-python-dev

KAPITEL 2

Software in der Virtuellen Machine

Die für den Kurs benötigte und oben beschriebene Software-Umgebung ist in der virtuellen Maschine *VM-DaA* (http://www.ekp.kit.edu/~quast/VM-DaA) enthalten. Diese virtuelle Maschine lässt sich leicht mit *VirtualBox* auf jedem Wirtssystem (Windows, Max OSX und allen Linux-Varianten) ausführen. Hinweise zur Installation und Anwendung von *VirtualBox* gibt es auf der Web-Seite des Herstellers (jetzt Oracle, s. http://www.virtualbox.org). Um weitere Funktionen, insbesondere den Zugriff auf USB-Ports des Wirtssystems zu erhalten, sollte auch die Erweiterung *Oracle VM VirtualBox Extension Pack* heruntergeladen und über den Menüpunkt *Preferences/Extensions* des graphischen Interfaces von VirtualBox installiert werden.

KAPITEL 3

Software unter Windows

Liste der zusätzlich benötigen Software under Windows (vers. 10):

- **empfohlen: `WinPython` (vers. 32 bit, 3.5.4)**

  (eine „Distribution“ mit den meisten gängigen Paketen):

  - `numpy`
  - `scipy`
  - `matplotlib`
  - (. . . sowie viele weitere)

- `iminuit` und `kafe`
- **kafe benötigt das Textsatz-Sytem `LaTeX`, empfohlen DANTE `texlive`** (identisch zur Version unter Linux)
- evtl. `root` / `pyroot` (32 bit, vers. 5.34.34)
- Eine „Entwicklungsumgebung“ für Projekte unter Python:
  - `IDLE` oder `spyder` (enthalten in `WinPython`)
  - **evtl. `pycharmEDU` (sehr mächtig, mit Index aller installierten Pakete),** enthält auch einen Einführungskurs zu Python (für didaktische Zwecke)

3.1 Bezugsquellen

Alle Pakete können prinzipiell direkt von den jeweiligen Projekt-Seiten heruntergeladen und installiert werden (Links s. Vorlesung).

Als bequemere (und schnellere) Möglichkeit für diesen Kurs wird eine gepackte, selbstextrahierende Datei `DAsoft.exe` bereit gestellt, die alle oben aufgeführten Komponenten enthält. Nach Entpacken z.B. in ein Verzeichnis `X:\<Pfad>\DAsoft` sind alle Komponenten unter Windows lauffähig. Das Installationsverzeichnis kann sich auch auf einem (schnellen) USB-Stick befinden. Die Datei `setpaths.bat` enthält die Befehle für die Windows-Eingabeaufforderung, um die notwendigen Pfade zu setzen (zur Erklärung siehe nächster Absatz).

3.2 Aufsetzen der Arbeitsumgebung

Damit alle Programme „sich gegenseitig kennen“, müssen sogenannte Umgebungsvariablen gesetzt werden. Dann kann man die Programme auch direkt über die Kommandozeile aufrufen, und Programmkomponenten der unteschiedlichen Pakete können von andern Paketen verwendet werden (`root` und `LaTeX` sind typische Beispiele).

Systemweit, d.h. für alle Benutzer und Programmpakete, geschieht das Setzen solcher Links über die grafische Oberfläche der Systemsteuereung von Windows.

Setzen von Pfaden mit der Windows-GUI

Rechtsklick Windows-Menü:

- -> Systemsteuerung
- -> System und Sicherheit
- -> System
- -> Erweiterte Systemeinstellungen
- -> Umgebungsvariablen
- **in Systemvariablen: Doppelklick Eintrag „Path“** „Neu“ anklicken und Pfade zu `Python`, `LaTeX` (und evlt. `root`) hinzufügen:
  - `X:\DAsoft\texlive\2017\bin\win32`
  - `X:\DAsoft\WinPython\python-3.5.4`
  - **`X:\DAsoft\root_v5.34.34\bin`** „`X:`“ bezeichnet dabei den Laufwerksnamen
- in Systemvariablen „Neu . . . “ anklicken und Variable `PYTHONPATH` hinzufügen, Eintrag für `root_<vers>\bin` anlegen:
  - `X:\DAsoft\root_v5.34.34\bin`

Pfade in der Windows-Eingabeaufforderung:

Die notwendigen Pfade lassen sich auch in der Windows-Eingabeaufforderung setzen. Wenn die Pfade bereits (wie oben angegeben) über die Systemsteuerung gesetzt sind, werden für die Eingabeaufforderung keine weiteren Pfadangaben benötigt.

Der Befehl zum dauerhaften Setzen von Pfaden, die auch für andere Anwendungen gelten, lautet:

`setx path "%path%;<weiterer_Pfad>"` (`%path%` enthält dabei die schon vorher gesetzten Pfade)

Das obige Beispiel zum Setzen des Pfades für LaTeX würde man in der Eingabeaufforderung so umsetzen:

```
setx path "%path%;X:\DAsoft\texlive\2015\bin\win32"
```

Für den Python-Pfad zu root:

```
setx pythonpath "%pythonpath%;X:\DAsoft\root_v5.34.34\bin"
```

Bitte beachten, dass die so gesetzten Pfade erst für neu gestartete Eingabeaufforderungen wirksam werden.

Symbolische Links im NTFS-Dateisystem

Praktisch sind auch „symbolische Links“ direkt im Dateisystem NTFS

(Achtung: das sind keine „Verknüpfungen“, wie sie mit der grafischen Oberfläche erzeugt werden !). Mit solchen Links, die man bei Bedarf erzeugen und wieder löschen kann, lassen sich generische Namen für die diversen Pakete verwenden, z.B. python -> python3.5.4

Vorgehensweise: Rechtsklick Windows-Menü „Eingabeaufforderung(Administrator)“:

- Symbolische Verknüpfungen hinzufügen:

  ```
  mklink /d <Verknüpfungsname> <Zielverzeichnis>
  ```

- Symbolische Verknüpfungen löschen:

  ```
  rmdir <Verknüpfungsname>
  ```

3.3 Installation des Anpassungs-Pakets kafe

Quellcode und Dolumentation zu *kafe* sind auf github (http://github.com/dsavoiu/kafe) verfügbar. *kafe* läuft unter *Python* 2.7 und 3.5 auf den Platformen *Linux*, *MS Windows (10)* und *Mac OS X*.

kafe hat Abhängigkeiten von externen Paketen, die alle, wie auf das Paket *kafe* selbst mit *pip* installiert werden können.

Zunächst *iminuit* oder, alternativ, das *root* Packet installiern. Eine bereits vorkompilierte Version für Windows, *iminuit-1.2-cp27-none-win32.whl* existiert und wird zunächst installiert:

```
pip install "iminuit<2"
```

dann:

```
pip install kafe
```

Das Entfernen von *kafe* geht ebenfalls über *pip*:

```
pip uninstall kafe
```

KAPITEL 4

Installation unter Mac OSX

Zuerst wird XCode installiert. Dieses Paket stellt u.a. Programme zum kompilieren der weiteren erforderlichen Pakete zur Verfügung. Dazu im Terminal folgenden Befehl ausfÃ¼hren und den Anweisungen auf dem Bildschirm folgen:

```
$ xcode-select --install
    [...]
$
```

Zum Installieren der verbleibenden erforderlichen Pakete empfehlen wir, einen Paketmanager zu nutzen (z.B. brew). Folgende Befehle dazu nacheinander in einem Terminal ausführen:

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/
↪master/install)"
    [...]
$ brew update
    [...]
$
```

Nun werden die Pakete ROOT und Python installiert (dieser Schritt kann eine Weile (~40 Minuten) dauern):

```
$ brew install root python
    [...]
$
```

Um ROOT verfügbar zu machen, müssen folgende Zeilen in die Datei `.bashrc` im Homeverzeichnis hinzugefügt werden:

```
thisroot=$(brew --prefix root)/bin/thisroot.sh
test -f $thisroot && source $thisroot
```

Desweiteren werden LaTeX und ein Editor zum Bearbeiten von `.tex`-Dateien installiert:

```
$ brew cask install mactex texshop
    [...]
$
```

Für die Datenanalyse werden nun noch die Python-Pakete `numpy` und `scipy`, und zum Plotten `matplotlib` mit dem Python-Paketmanager `pip` installiert:

```
$ pip3 install numpy scipy matplotlib
    [...]
$
```

Unter Umständen ist es nötig, das sogenannte Backendend von `matplotlib` auf `TKAgg` umzustellen, falls Fenster mit Plots nicht dargestellt werden. Das kann entweder mit

```
import matplotlib
matplotlib.use('TKAgg')
```

mit jedem `matplotlib`-import geschehen, oder es wird `backend : TkAgg` an das Ende der Datei `$USER/.matplolib/matplotlibrc` angefügt. Hier können auch weitere globale matplotlib-Einstellungen vorgenommen werden, siehe die matplotlib Dokumentation.

Zu guter Letzt werden noch iminuit und kafe (Karlsruhe Fitting Environment vers. 2) für Minimierungs-/Fittingprobleme installiert

```
$ pip3 install 'iminuit<2' kafe2
    [...]
$
```