

# Willkommen beim Sage-Installationsführer

8. April 2010

Dieses Dokument ist eine abgeleitete Übersetzung erstellt von Bernhard Blöchl von „Welcome to the Sage Installation Guide“ (<http://sagemath.org/doc/installation/index.html>) © Copyright 2005–2010, The Sage Development Team, licensed under a Creative Commons Attribution-Share Alike 3.0 License <http://creativecommons.org/licenses/by-sa/3.0/.deed.de>.

(This work is a derivative work, a translation prepared by Bernhard Bloechl from „Welcome to the Sage Installation Guide“ (<http://sagemath.org/doc/installation/index.html>) © Copyright 2005–2010, The Sage Development Team, licensed under a Creative Commons Attribution-Share Alike 3.0 License <http://creativecommons.org/licenses/by-sa/3.0/>.)

Um Fehler des Documents zu melden oder für Anregungen, senden Sie bitte ein Mail an *sagemath-at-arcor.de*. (Keine Hilfe für Installationsfragen oder zu Sage! Für solche Fragen nutzen Sie bitte eine der Möglichkeiten auf <http://www.sagemath.org/development-groups.html>)

Das vorliegende Dokument gibt eine kurze Anleitung für die Erstinstallation von Sage. Einmal installiert, können Sie auf eine neuere Version von Sage mit der Eingabe „sage -upgrade“ upgraden oder Sie geben „sage -h“ ein für alle verfügbaren Optionen im Zusammenhang mit der Installation der Sage-Pakete.

## Inhaltsverzeichnis

<b>1</b>	<b>Einführung</b>	<b>2</b>
<b>2</b>	<b>Kurzführer Download und Installation</b>	<b>5</b>
2.1	Fehlerbehebung . . . . .	5
<b>3</b>	<b>Installieren von Binärversionen</b>	<b>5</b>
3.1	Linux und OS X . . . . .	5
3.2	Microsoft Windows . . . . .	7

<b>4</b>	<b>Installation aus dem Quellcode</b>	<b>7</b>
4.1	Fortran . . . . .	9
4.2	Schritte der Installation aus der Quelle . . . . .	10
4.3	Installation in einer Multiuser-Umgebung . . . . .	14
4.4	Besondere Hinweise . . . . .	15
<b>5</b>	<b>SageTeX mit TeX bekannt machen</b>	<b>16</b>
5.1	SageTeX-Dokumentation . . . . .	17
5.2	SageTeX und TeXLive . . . . .	18
<b>6</b>	<b>Desktop-Startsymbol</b>	<b>18</b>
<b>7</b>	<b>Dokumentation</b>	<b>19</b>

## 1 Einführung

Sie können Sage entweder aus einer bereits kompilierten Binärdatei von einem Tarball für Ihre Betriebssystemversion installieren oder aber aus dem Quellcode. Die Installation aus dem Binärkode heraus ist am schnellsten und hat die wenigsten Voraussetzungen. Die Installation aus der Quelle erlaubt die Verwendung der neuesten Version von Sage und stellt sicher, dass Sie gegebenenfalls den Quellcode von Sage modifizieren und re-kompilieren können. Die Installation aus dem Quellcode ist einfacher, als Sie das von anderer Software kennen, weil viele Tests ausgeführt werden, die sicherstellen, dass Sage und alle Komponenten erfolgreich ohne Nutzerinteraktion auf einer breiten Palette von Computern kompiliert werden. Es ist jedenfalls einfacher, Sage vollständig aus der Quelle zu kompilieren, als wenn Sie die Pakete, die Bestandteil von Sage sind (z.B. Singular) einzeln kompilieren.

Die Sage-Distribution enthält die meisten Programme, von denen Sage abhängt. Eine unvollständige Liste ist unten gegeben. Alle diese Programme stehen unter einer GPL-kompatiblen Lizenz (siehe die Datei COPYING.txt im Sage-Heimverzeichnis für Details).

Hier ist eine Liste einiger Software, die in Sage enthalten ist:

- atlas: das Projekt ATLAS (Automatically Tuned Linear Algebra Software)
- bzip2: die bzip2-Kompressionsbibliothek
- ecl: der common lisp-Interpreter
- cython: die Cython-Programmiersprache, eine auf Pyrex basierende Sprache, mit der einfach C-Erweiterungen für Python geschrieben werden können.
- eclib: John Cremonas Programme für die Berechnung und das Rechnen mit elliptischen Kurven, die über rationale Zahlen definiert sind.
- ecm: Methode der elliptischen Kurven für die Faktorisierung von Integern
- flint: schnelle Bibliothek für die Zahlentheorie

## 1 Einführung

- fortran: die Programmiersprache Fortran
- GAP: Ein System für numerische diskrete Algebra
- genus2reduction: Reduzierung der Information mittels Kurven vom Geschlecht 2
- gfan: Berechnung von Groebner-Fächern und Torus-Varianten
- ghmm: Modellbibliothek für versteckte Markov-Prozesse
- givaro: eine C++ -Bibliothek für arithmetische und algebraische Berechnungen
- gmp-mpir: MPIR ist eine OpenSource Bibliothek für höhere numerische Genauigkeit, abgeleitet von GMP (GNU multiprecision library)
- gsl: GNU Scientific Library ist eine numerische Bibliothek für C- und C++-Programme
- ipython: eine erweiterte Python-Shell für effiziente, interaktive Eingabe, eine Bibliothek zur Programmierung interaktiver Umgebung basierend auf Python sowie ein System für verteilte und parallelisierte Programmierung
- jmol: ein Java „Molecular Viewer“ für dreidimensionale chemische Strukturen
- jsmath: stellt Mathematik für HTML zur Verfügung
- lapack: eine Bibliothek von Fortran 77 Subroutinen für die am häufigsten auftretenden Problemen der linearen Algebra
- lcalc: Rubinsteins L-functions-Rechner
- libfpLLL: enthält verschiedene Implementierungen für den Gleitkomma „LLL-Reduktionsalgorithmus“, der unterschiedliche Geschwindigkeits- und Zuverlässigkeitsverhältnisse bereitstellt
- libm4ri: Bibliothek für Matrizen-Multiplikation, -Reduktion und Inversion über  $GF(2)$
- linbox: C++ Musterbibliothek für genaue, hochperformante Berechnung der linearen Algebra
- matplotlib: eine Python 2-D-Bibliothek für Plotterstellungen
- maxima: symbolische Algebra und Differentialrechnung
- mercurial: ein Quellcode-Managementsystem zur Verwaltung großer verteilter Projekte
- mpfi: eine C-Bibliothek für Arithmetik auf der Grundlage von MPFR (Multi-precision Intervals) und GMP 1
- mpfr: eine C-Bibliothek für wählbare Genauigkeit von Gleitkommaoperationen mit korrekter Rundung

## 1 Einführung

- networkx: ein Python-Paket für die Erstellung, die Bearbeitung und die Analyse von Strukturen, Dynamik und Funktionen komplexer Netze.
- NTL: C++ Bibliothek für die Zahlentheorie
- numpy: numerische lineare Algebra und andere numerische Rechenverfahren für Python
- palp: ein Paket zur Analyse von „Lattice Polytopes“
- pari: PARI, Bibliothek für Zahlentheorie
- pexpect: Python-Expect (für die Fernsteuerung anderer Systeme)
- polybori: Stellt High-level-Datentypen für boolesche Polynomiale und Monomiale, Exponentialvektoren sowie für die zugrunde liegenden polynomiale Ringe und Untermengen der Potentialmenge der booleschen Variablen
- pynac: eine modifizierte Version von GiNaC (eine C++ Bibliothek für symbolische mathematische Berechnungen) die die Abhängigkeit von Python von CLN eliminiert
- Python: Die Python-Programmiersprache
- R: eine Sprache und eine Umgebung für statistische Berechnungen und Grafiken
- readline: die GNU-Bibliothek Readline line editor
- scipy: wissenschaftliches Tool für Python
- singular: Polynomiale Berechnungen für die algebraische Geometrie u.s.w.
- symmetrica: Routinen für Berechnungen im Bereich der Darstellungstheorie für klassische und symmetrische Gruppen und verwandte Felder
- sympow: Symmetrische L-Potenzfunktionen und modulare Ordnung
- sympy: Python-Bibliothek für symbolische Mathematik
- tachyon: Tachyon(tm) parallele Multiprozessor-Ray-Tracing-Software
- termcap: Bibliothek für Terminalanzeigen
- Twisted: Networking-Framework
- zlib: zlib-Kompressionsbibliothek
- zn\_poly: C-Bibliothek für polynomiale Arithmetik in  $\mathbb{Z}/n\mathbb{Z}[x]$
- ZODB: Zope-Objektdatenbank

## 2 Kurzfürer Download und Installation

Sie sind unsicher, was Sie downloaden sollen? Dieser Kurzfürer sagt es Ihnen.

- Bestimmen Sie Ihr Betriebssystem (Windows, Linux, Mac OS X, etc.).
- Bestimmen Sie den Typ Ihrer CPU (32-bit, 64-bit oder "atom" für Linux und Intel oder PowerPC für Mac OS X).
- Wollen Sie eine Binär- oder Quellcode-Distribution? Selbst wenn Sie entwickeln wollen, ist eine Binärdistribution von Sage geeignet. Die Quellcode-Distribution ist hauptsächlich erforderlich, wenn Sie den Quellcode der Pakete analysieren wollen, die gewöhnlich als SPKs bezeichnet werden.
- Wenn verfügbar, wählen Sie die passende Binärdistribution von einem der „download mirrors“. Eine Liste der Spiegelserver ist unter <http://www.sagemath.org/mirrors.html> verfügbar.
- Befolgen Sie die Anleitung für die Binärinstallation (<http://www.sagemath.org/doc/installation/binary.html>), um eine aktuelle vorkompilierte Version von Sage zu installieren. Die Anleitung für die Installation von der Quelle (<http://www.sagemath.org/doc/installation/source.html>) bietet ausführlichere Informationen zur Kompilation von Sage aus dem Quellcode.

### 2.1 Fehlerbehebung

- Wenn für Ihr System keine Binärversion verfügbar ist, laden Sie die Quellcode-Version herunter. Sage kompiliert auf einer großen Zahl verschiedener Systeme, jedoch nicht auf allen.
- Wenn Sie eine Binärversion von Sage heruntergeladen haben, könnte beim Starten von Sage eine Meldung eines „illegal instruction error“ erscheinen. In diesem Fall ist eine Lösung auf der FAQ-wiki-Seite <http://wiki.sagemath.org/faq#Otherquestions> gegeben.
- Für Fragen verwenden Sie die „sage-support mailing list“. Diese Mailliste wird auch als „sage-support Google group“ bezeichnet (<http://groups.google.com/group/sage-support>).

## 3 Installieren von Binärversionen

### 3.1 Linux und OS X

Die Installation eines kompilierten Binär-Tarballs scheint der einfachste und schnellste Weg zu sein, um Sage zu installieren. Das ist jedoch nicht ohne Vorbehalt richtig. Beachten Sie, dass Sage selbst eine Programmierumgebung ist, weshalb die Erstellung aus dem

### 3 Installieren von Binärversionen

Quellcode langfristig die maximale Flexibilität sicher stellt. Dennoch steht Sage auch als Binärversion zum Download zur Verfügung.

Voraussetzungen: Sie haben einen Rechner mit mindestens 2 GB freiem Speicher und dem Betriebssystem Linux (32-bit oder 64-bit) oder OS X (10.5.x).

Empfehlung: Es ist sehr empfehlenswert, dass Sie  $\text{\LaTeX}$  installiert haben.

Laden Sie den aktuellen Binär-Tarball von <http://www.sagemath.org/download.html> herunter. Dieser kann z. B. als „sage-x.y.z-x86\_64-Linux.tgz“ bezeichnet sein. Entpacken Sie den Tarball in einem Verzeichnis, für das Sie Lese- und Schreiberlaubnis haben.

```
tar zxvf sage-x.y.z-x86_64-Linux.tgz
```

Das so entstehende Verzeichnis „sage-x.y.z-x86\_64-Linux“ können Sie beliebig verschieben und Sie können Sage von dort starten. Sie können ebenso die Datei „sage“ des Verzeichnisses in ein beliebiges Verzeichnis kopieren, z. B. nach `/usr/local/bin/`. In diesem Fall müssen Sie jedoch anschließend den obersten Eintrag

```
SAGE_ROOT="...."
```

editieren und die Punkte durch das Sage-Verzeichnis

```
SAGE_ROOT=sage-x.y.z-x86_64-Linux
```

ersetzen. Wenn der Eintrag `/usr/local/bin/` sich im Pfad Ihres Rechners befindet., können Sie Sage von überall durch einfache Eingabe von „sage“ in einer Eingabezeile starten. Ein anderer Ansatz ist die Erstellung eines symbolischen Links von `sage-x.y.z-x86_64-Linux` nach beispielsweise `/usr/local/share/sage`:

```
ln -s /.../path_to/.../sage-x.y.z-x86_64-Linux /usr/local/share/sage
```

Ergänzen Sie dann `$PATH` mit `/usr/local/share/sage`. Danach können Sie ebenfalls Sage durch die Eingabe von „sage“ in einer Eingabezeile starten. Wenn Sie eine andere oder eine neue Version von Sage installieren, müssen Sie den alten Link löschen und einen neuen erstellen, der auf das Verzeichnis `/usr/local/share/sage` zeigt.

Schlussendlich können Sie auch diese zwei Ansätze verbinden, sage nach `/usr/local/bin/` kopieren, einen Link von `sage-x.y.z-x86_64-Linux` zu `/usr/local/share/sage` erstellen und die Zeile der Datei `/usr/local/bin/sage`:

```
SAGE_ROOT="...."
```

```
in
```

```
SAGE_ROOT="/usr/local/share/sage"
```

ändern. Wenn Sie eine neue Version von Sage installieren, löschen Sie den alten Link und erstellen Sie einen neuen – die Datei `/usr/local/bin/sage` müssen Sie nicht ändern.

Wenn Sie Sage das erste Mal starten und immer wenn Sie das Sage-Verzeichnis verschieben oder wie oben beschrieben einen neuen Link erstellen, erscheint eine Meldung:

```
The Sage install tree may have moved.
Regenerating Python.pyo and .pyc files that hardcode the install PATH
(please wait at most a few minutes)...
Do not interrupt this.
```

Die aktuelle Distribution enthält `.dmg`-Dateien für OS X 10.4.x und 10.5.x. Aber Sage soll eine eigenständige Anwendung werden (native application). Daran wird gearbeitet, Hilfe ist immer willkommen.

### 3.2 Microsoft Windows

Die beste Art Sage unter Windows zu installieren ist die Installation in einer VirtualBox für Windows und darin die „VirtualBox distribution of Sage“ zu installieren. Für weitere Anweisungen zur Installation von Sage auf Windows folgen Sie dem Link zu dieser URL (<http://www.sagemath.org/download-windows.html>). Lesen Sie auf jeden Fall die Datei README.txt.

## 4 Installation aus dem Quellcode

Um Sage aus dem Quellcode zu erstellen, ist etwas mehr Vertrautheit mit dem Computer erforderlich. Sind aber alle Voraussetzungen zur Kompilierung erfüllt, ist der Prozess einfach. Ihr Computer wird eine Weile beschäftigt sein, um Sage aus dem Quellcode zu kompilieren, aber Sie müssen das nicht überwachen. Wenn Sie Sage aus der Quelle kompilieren, können Sie sicher sein, die neueste Version von Sage zu haben, mit der Sie alle Bestandteile des Programms ändern können, von dem Sage abhängt. Sie können Sage auch rekompilieren.

Zum Zeitpunkt der Texterstellung für diese Anleitung (Install Guide v4.3.4, 2010) wird Sage auf Linux (32-bit x86, 64-bit x86-64, IA64 oder 32-bit PPC) und OS X (10.4, 10.5, 10.6, PPC oder x86, nur 32-bit) unterstützt (Siehe <http://wiki.sagemath.org/SupportedPlatforms> für die neuesten Informationen.)

Solaris? FreeBSD? OS X 10.5 im 64bit-Modus?: Die vollständige Kompilation von Sage ist zum aktuellen Zeitpunkt nicht auf Solaris oder \*BSD möglich. Der größte Teil von Sage kann auf Solaris kompiliert werden und die weiteren Bestandteile können mit Standardpaketen ergänzt werden; E-mailen Sie an sage-devel <http://groups.google.com/group/sage-devel>, wenn Sie Sage auf Solaris dringend benötigen. Die volle Unterstützung von Solaris ist geplant – die Arbeit daran wird fortgesetzt.

Sage auf FreeBSD: Die Binärversionen können unter der Linux-Emulation ausgeführt werden. Das Sage-Entwicklungsteam Team arbeitet an einem vollen „native port“ und die Zahl der Änderungen ist im Vergleich zu anderen Ports relativ klein.

Das Sage-Entwicklungsteam hofft, OS X 10.5 im 64-bit Mode für die nächste Sage-Version unterstützen zu können. Einige Informationen zum Kompilieren von Sage auf OS X 10.5 im 64-bit Mode finden Sie unter: <http://mvngu.wordpress.com/2009/09/02/compile-sage-4-1-in-64-bit-mode-on-os-x-10-5-8/>

Annahmen: Sie haben einen Rechner mit installiertem Linux (32-bit or 64-bit), Mac OS X 10.4, 10.5, oder 10.6 mit Xcode und 2 GB freien Plattenplatz.

Im Einzelnen müssen unter Linux die folgenden Standard-Entwicklungstools für Kommandozeileineingabe installiert sein. (Unter OS X sind alle mit XCode installiert.):

```
gcc
g++
gfortran
make
m4
perl
ranlib
tar
```

## 4 Installation aus dem Quellcode

readline mit den development headers  
ssh-keygen wird benötigt, um Notebooks im sicheren Mode zu benutzen  
latex wird empfohlen, ist aber nicht unbedingt erforderlich

Um festzustellen, ob m4 installiert ist, geben Sie in der Eingabezeile „which m4“ ein. Wenn kein Antworttext (oder eine Fehlermeldung) erfolgt, dann ist m4 nicht installiert. Außerdem wird die Installation von L<sup>A</sup>T<sub>E</sub>X sehr empfohlen, sie ist aber nicht unbedingt erforderlich. Wenn Sie ssh-keygen nicht installiert haben, können Sie Ihr Notebook nicht im sicheren Mode betreiben. Um Ihr Notebook im unsicheren Mode zu betreiben, starten Sie Ihr Notebook mit der Eingabe „notebook(secure=False)“ anstelle von „notebook()“.

Stellen Sie sicher, dass Sie unter OS X zumindest die Version 2.4 von Xcode haben, gcc -v sollte zumindest 5363 ausgeben. Ist dies nicht der Fall, laden Sie von <http://developer.apple.com/> nach der Registrierung das freie XCode-Paket herunter. Lediglich OS X  $\geq 10.4$  wird unterstützt. Damit haben Sie alle oben gegebenen Befehle zur Verfügung.

Bei einem Debian-basierten System (z. B. Ubuntu), ist ranlib im Package binutils inkludiert. Auf einem neu installierten Ubuntu-System (getestet mit Ubuntu 9.04), können Sie die oben gegebenen Befehle wie folgt installieren:

```
sudo apt-get install build-essential m4 gfortran
```

Es wird empfohlen, die readline packages und die zugehörigen development headers zu installieren. Diese Pakete erleichtern die Arbeit mit der Sage Eingabezeile, da sie Texteditorfunktionen für die Anweisungen der Eingabezeile zur Verfügung stellen. Auf einem Debian- oder Ubuntu-System nutzen Sie die folgende Eingabe um die readline library und die zugehörigen development headers zu installieren:

```
sudo apt-get install readline-common libreadline-dev
```

Das L<sup>A</sup>T<sub>E</sub>X-Paket und ein PDF-Leser sind optional, die Installation erfolgt mit:

```
sudo apt-get install texlive xpdf evince xdvi
```

(Es muss die GNU-Version von make installiert sein. Beispielsweise erstellt Sage kein Build auf FreeBSD, wenn keine GNU-Version von make (unter genau dem Namen make) installiert ist.

Einige Teile von Sage sind in Python geschrieben, dennoch muss Python nicht auf Ihrem Computer vorinstalliert werden, weil die Sage-Installation alles mitbringt, was Sie benötigen.

Bei der Ausführung des Installationsprogramms wird geprüft, ob alle oben aufgeführten benötigten Programme vorhanden sind und es erfolgt eine Meldung darüber, welche vermisst werden.

- Wenn Sie die Tcl/Tk-Bibliotheken in Sage nutzen wollen, bereiten Sie das vor, indem Sie die Tcl/Tk-Entwicklungsbibliothekens zusätzlich zu der Tck/Tk-Basisinstallation installieren. Das Python von Sage erkennt Tcl/Tk automatisch, wenn es installiert ist.

Auf Ubuntu erreichen Sie das durch die folgende Anweisung:

```
sudo apt-get install tk8.5-dev # or the latest version available
```



## 4 Installation aus dem Quellcode

Danach können Sie Sage installieren und Sages Python erkennt automatisch die Systeminstallation von Tcl/Tk. Wenn Sie das vergessen haben und Sage zuerst installiert haben, ist nicht alles verloren. Geben Sie dann nach der oben beschriebenen Installation der Tcl/Tk-Entwicklungsbibliotheken die folgende Anweisung ein:

```
sage -f python-4.3.4.p8 # or the latest version available
```

Wenn danach die Befehle zur Installation keine ImportError-Fehlermeldung ausgeben, hat das funktioniert:

```
sage: import _tkinter
sage: import Tkinter
```

- Sage wird aktuell mit GCC Version 4.3.x entwickelt, kompiliert aber auch mit anderen GCC-Versionen der 4.x-Serie einwandfrei, kompiliert aber nicht mit älteren GCC-Versionen. Sollten Sie an einer Version für Suns CC-Compiler interessiert sein, schicken Sie bitte eine E-Mail an die sage-devel mailing list, auch bekannt als sage-devel Google group unter <http://groups.google.com/group/sage-devel>
- Perl wird benötigt, weil die NTL- und PARI-Konfigurationsskripts in Perl geschrieben sind.

Nach dem Entpacken des Sage-Tarballs enthält das Unterverzeichnis spkg die Quellcodes für alles, was Sage benötigt. Die Entwickler legen Wert darauf, dass die gesamte benötigte Software in der Sage-Distribution enthalten ist, so dass Sie sich nicht selbst um Download und Installation einzelner Pakete kümmern müssen (wie Z. B. GAP).

Bekannte Probleme mit verschiedenen Linux-Systemen sind:

- Kein Build mit gcc 4.3.0, die Behebung ist in Arbeit.
- Verschieben des Builds nach der Kompilation bricht die "PARI Galois fields database", da die Pfade offensichtlich im PARI-Binary fix sind. PARI hat also eine starre, vorgegebene Verzeichnisstruktur. (Hilfe, das zu korrigieren ist erwünscht!)

### 4.1 Fortran

Unter Linux und Solaris ist ein Fortran-Compiler erforderlich, um Sage aus der Quelle zu kompilieren. Wenn Sie Fortran auf einer Plattform benutzen, für die Sage keine g95-Binarversion anbietet, müssen Sie systemweit gFortran benutzen. Solaris 10 beispielsweise wird ohne Fortran-Binärdateien ausgeliefert. Sie müssen dem Sage-Build-Prozess den Ort des Fortran-Compilers und der Bibliotheken explizit mitteilen. Das tun Sie so:

```
export SAGE_FORTRAN=/exact/path/to/gfortran
export SAGE_FORTRAN_LIB=/path/to/fortran/libs/libgfortran.so
```

Beachten Sie, dass die Umgebungsvariable SAGE\_FORTRAN lediglich das Fortran-Sage-Paket betrifft, das sonst als Fortran spkg bezeichnet wird. Darüber hinaus soll diese Variable andere spkgs nicht beeinflussen, die Fortran benutzen. Beispielsweise nutzt das Lapack spkg Fortran, aber der Kompilationsprozess von Lapack soll die Umgebungsvariable SAGE\_FORTRAN ignorieren. Die SAGE\_FORTRAN-Umgebungsvariable bedeutet

## 4 Installation aus dem Quellcode

nicht: „Kompiliere alle spkg, die Fortran nutzen, mit diesem Fortran“. Vielmehr bedeutet sie: „Nutze das `sage_fortran` script der `SAGE_FORTRAN`-Umgebungsvariablen, damit der Fortran-Compiler für die Installation des Fortran spkg genutzt wird“.

Auf Mac OS X benötigen Sie keinen Fortran-Compiler. Die Sage-Quelcodedistribution benötigt keinen Fortran-Compiler auf diesem System. Für Mac OS X wird die Sage-Quelcodedistribution mit einem Fortran-Compiler für Mac OS X ausgeliefert. Dieser Fortran-Compiler wird benutzt, selbst wenn Sie einen anderen Fortran-Compiler in der Umgebungsvariablen `SAGE_FORTRAN` definieren.

Auf den Plattformen AIX, HP-UX und Solaris, für die sowohl 32- und 64-bit Builds verfügbar sind, muss die Variable im Pfad zur Bibliothek `SAGE_FORTRAN_LIB` auf die 32-bit-Bibliothek verweisen, wenn Sie Sage für 32-bit kompilieren, und auf die 64-bit-Bibliothek, wenn Sie Sage für 64-bit kompilieren. Unter Solaris können die Variablen `SAGE_FORTRAN` und `SAGE_FORTRAN_LIB` beispielsweise wie folgt gesetzt werden (Sie sollten das prüfen):

```
# SPARC and x86
SAGE_FORTRAN=/path/to/gcc/install/directory/bin/gfortran
# 32-bit SPARC
SAGE_FORTRAN_LIB=/path/to/gcc/install/directory/lib/libgfortran.so
# 64-bit SPARC
SAGE_FORTRAN_LIB=/path/to/gcc/install/directory/lib/sparcv9/libgfortran.so
# 32-bit x86
SAGE_FORTRAN_LIB=/path/to/gcc/install/directory/lib/libgfortran.so
# 64-bit x64
SAGE_FORTRAN_LIB=/path/to/gcc/install/directory/lib/amd64/libgfortran.so
```

### 4.2 Schritte der Installation aus der Quelle

Die Installation aus der Quelle ist (potentiell) sehr einfach, weil die Distribution essentiell alles enthält, was Sage benötigt.

Stellen Sie sicher, dass keine Leerzeichen in den Verzeichnisnamen sind, unter denen Sie kompilieren. Zwar wird der Start aus einem Verzeichnis mit Leerzeichen im Bezeichner unterstützt, dieser wird aber nicht empfohlen, da das Kompilieren aus einem solchen Verzeichnis nicht möglich, weil einige Komponenten nicht kompilieren, wenn sich Leerzeichen im Pfad befinden.

1. Rufen Sie <http://www.sagemath.org/download-source.html> auf, wählen Sie einen Mirror und laden Sie die Datei `sage-*.tar` herunter.

Dieser Tarball enthält den Quellcode für Sage und den Quellcode für alle Programme, von denen Sage abhängt. Speichern Sie den Download in einem Unterverzeichnis Ihres Heimatverzeichnisses, in das Sie Sage installieren wollen. Beachten Sie, dass diese Datei nicht komprimiert ist, es handelt sich um einen einfachen Tarball, (der allerdings voll mit komprimierten Dateien ist).

2. Extrahieren Sie:

```
tar xvf sage-x.y.z.tar
```

3. Das erzeugt ein Verzeichnis `sage-x.y.z`.

## 4 Installation aus dem Quellcode

4. Wechseln Sie in das Verzeichnis

```
cd sage-x.y.z
```

Das ist das Heimatverzeichnis von Sage, das als `SAGE_ROOT` oder als „top level Sage directory“ bezeichnet wird.

5. Optional (aber sehr empfohlen): Lesen Sie die Datei `README.txt`.

6. Geben Sie ein:

```
make
```

Damit werden Sage und alle Abhängigkeiten kompiliert. Sie sollen und brauchen nicht als Administrator oder als Root eingeloggt sein, weil keine der Dateien außerhalb des Verzeichnisses `sage-x.y.z` verändert werden.

Diese Anweisung führt die üblichen Schritte für jedes Paket aus, legt aber alle Kompilate im lokalen Build-Verzeichnisbaum ab. Das kann je nach Maschine bis zu einer Stunde dauern. Abhängig von der Architektur des Systems (z. B., Celeron, Pentium Mobile, Pentium 4, usw.), kann es bis zu drei Stunden dauern, um Sage aus der Quelle zu erstellen. Wenn der Build erfolgreich war, erscheint das Wort `ERROR` nicht in den letzten 3-4 Zeilen Ihrer Ausgabe.

Es gibt eine Ausnahme – das Verzeichnis `.ipythonrc` wird in Ihrem HOME-Verzeichnis erstellt, sofern es nicht vorhanden ist (<http://www.sagemath.org/doc/installation/source.html#id1>).

Die Verzeichnisstruktur ist NICHT absolut kodiert. Es ist möglich, das Verzeichnis zu verschieben oder umzubenennen. (Wenn das nicht funktioniert, ist das ein Fehler – unglücklicherweise existiert ein Fehler, der bisher nicht behoben ist und der PARI betrifft, das den Ort der Dateien „galois data“ festschreibt. Korrektur ist erwünscht!)

Nachdem Sage kompiliert wurde, können Sie die gesamte installierte Verzeichnisstruktur zu `/usr/local` kopieren. Sie können ebenso `sage-*/sage script` zu `/usr/local/bin/` kopieren und `ROOT="...."` am Beginn dieser Datei editieren.

1. Um Sage zu starten, wechseln Sie in das Sage-Verzeichnis und geben Sie ein:

```
./sage
```

Sie sollten das Sage-Prompt sehen, das etwa so aussieht (Der erste Start kann einige Sekunden dauern):

```
$ sage
```

---

```
| SAGE Version 3.0.5, Release Date: 2008-07-11 |  
| Type notebook() for the GUI, and license() for information. |
```

---

```
sage:
```

Wenn Sage erfolgreich startet, heißt das, dass die meisten der Komponenten erfolgreich kompiliert wurden. Wenn das Sage-Prompt aber nicht erscheint (und Sie

#### 4 Installation aus dem Quellcode

eine ausführliche Fehlermeldung, ein „traceback“, erhalten), berichten Sie bitte das Problem z. B. an <http://groups.google.com/group/sage-support> und fügen Sie Ihrer E-Mail bitte die Datei `install.log` bei. Hilfreich ist außerdem die Mitteilung des Betriebssystems und die Versionsnummer (und das Datum) der Sage-Kopie, die Sie nutzen. (Es gibt keine bestimmte Form für Fehlerberichte – senden Sie, wir schätzen jeden Bericht.) Versuchen Sie nach dem Start eine Eingabe:

```
sage: 2 + 2
4
```

Versuchen Sie etwas kompliziertere Eingaben, etwa eine, die die PARI C-Bibliothek benutzt:

```
sage: factor(2005)
5401
```

Prüfen Sie, ob die Schnittstellen zu Gap, Singular, Maxima and PARI/GP funktionieren:

```
sage: gap('2+2')
4
sage: gp('2+2')
4
sage: maxima('2+2')
4
sage: singular('2+2')
4
sage: pari('2+2')
4
```

(Hinweis für GAP-Kenner: Sage erstellt bei der Installation automatisch einen GAP “workspace”, deshalb ist die Antwortzeit für GAP-Anweisungen relativ schnell. Hinweise für GP/PARI-Kenner: Eine `gp`-Anweisung erzeugt ein Objekt im GP-Interpreter und eine `pari`-Anweisung ein Objekt in der PARI C-Bibliothek.)

Versuchen Sie GAP, Singular oder GAP aus Sage:

```
sage: gap_console()
GAP4, Version: 4.4.6 of 02-Sep-2005, x86_64-unknown-linux-gnu-gcc
gap> 2+2;
4
[Strg+d]
```

```
sage: gp_console()
...
[Strg+d]
```

```
sage: singular_console()
          SINGULAR
A Computer Algebra System for Polynomial Computations  / Development
                                                         / version 3-0-1
0 <
                                                         \ October 2005
          by: G.-M. Greuel, G. Pfister, H. Schoenemann
FB Mathematik der Universitaet, D-67653 Kaiserslautern
// ** executing /usr/local/sage/sage-0.8.2/bin/LIB/.singularrc
[ctrl-d]
> Auf Wiedersehen.
sage:
```

## 4 Installation aus dem Quellcode

- Optional: Prüfen Sie die Schnittstelle zu jeder in Sage nicht enthaltenen Software, die Sie verfügbar haben. Beachten Sie, dass jeder Aufruf eines externen Programms einen speziellen Namen hat: Mathematica wird mit `math`, Maple mit `maple` usw. aufgerufen. Der einfachste Weg, diesen Namen zu ändern oder eine andere Anpassung vorzunehmen, ist die Erstellung eines Umleitungsskripts in `$$SAGE_ROOT/local/bin`. Sage schreibt dieses Verzeichnis an die erste Stelle der Pfadangabe Ihres Rechners, sodass Ihr Skript absolute Pfade nutzen muss, um einen Aufruf auf sich selbst zu vermeiden. Außerdem muss Ihr Skript `$$*` nutzen, um alle Argumente zu übergeben. Ein Maple-Skript kann wie folgt aussehen:

```
#!/bin/sh
/etc/maple10.2/maple.tty $$*
```

- Optional: Verschiedene Möglichkeiten, um Sage leichter handhabbar zu machen:
  - Kopieren Sie `$$SAGE_ROOT/sage` in den Pfad Ihrer Maschine. Wenn Sie dies tun, müssen Sie auch die Zeile mit den Punkten (`....`) im oberen Teil des Skripts editieren.
  - KDE-Nutzer müssen ein `bash`-Skript, z. B. mit dem Namen „sage“ mit den folgenden Zeilen erstellen:

```
#!/bin/bash
konsole -T "sage" -e <SAGE_ROOT>/sage
```

Nachdem Sie dieses ausführbar gemacht haben (`chmod a+x sage`), setzen Sie es in Ihren Pfad. (Beachten Sie, dass Sie `$$SAGE_ROOT` ändern müssen, wie das oben bereits beschreiben wurde!) Sie können aber auch ein KDE-Startsymbol auf Ihrem Desktop mit dieser Anweisung versehen. Klicken Sie dazu mit der rechten Maustaste auf das Symbol, wählen Sie Einstellungen und tragen Sie im geeigneten Feld den Startbefehl ein.

- Nutzer der `bash`-Shell geben „`echo $PATH`“ in die Eingabezeile ein und kopieren den `sage`-Pfad mit „`cp sage <your-path-dir>`“ in eines der Verzeichnisse oder fügen dieses Binärverzeichnis der `PATH`-Variablen hinzu. Wenn Sie die `bash`-Shell nutzen, fügen Sie z. B. die folgenden Zeilen Ihrer Datei `„.bashrc“` hinzu (sofern diese existiert; wenn nicht, erstellen Sie die Datei):

```
PATH="<sage-home-dir>/bin":$PATH
export PATH
```

Danach loggen Sie sich aus und wieder ein, damit die Änderung aktiv wird. Geben Sie dann „sage“ in der Eingabezeile Ihrer Konsole ein und Sage sollte starten.

- Unter Linux und OS X können Sie ein „Alias“ zu `$$SAGE_ROOT/sage` erstellen. Z. B. fügen Sie etwa eine Zeile wie die folgende in Ihre Datei `„.bashrc“`:

```
alias 'sage'='/home/username/sage-3.1.2/sage'
```

Danach schließen Sie Ihr Terminal und starten es neu. Geben Sie dann „sage“ in der Eingabezeile Ihrer Konsole ein. Sage sollte nun starten.

## 4 Installation aus dem Quellcode

4. Optional: Testen Sie die Installation mit der Eingabe „./sage -testall“ in der Eingabezeile. Damit werden die meisten Beispiele im Quellcode ausgeführt und es wird sichergestellt, dass diese so wie erwartet ausgeführt werden. Um alle Beispiele auszuführen, nutzen Sie „./sage -testall -optional -long“: Damit werden alle Beispiele ausgeführt und auch die Beispiele, die von optionalen Paketen und Software abhängen, wie z. B. Mathematica oder Magma. Dies dauert etwas länger. Die Ausführung einiger (optionaler) Beispiele wird möglicherweise scheitern, weil sie voraussetzen, dass eine Datenbank installiert ist. Alternativ können Sie im Sage-Rootverzeichnis (\$SAGE\_ROOT) „make test“ eingeben, um alle Standardtests auszuführen. Das kann zwischen etwa 30 Minuten und einer Stunde oder länger dauern.
5. Optional: Das Verzeichnis „spkg/build“ enthält „intermediate code“, der benutzt wird, um Sage lauffähig zu erstellen. Geben Sie „make clean“ ein, um diesen und einige überflüssige Verzeichnisse zu entfernen (z. B. spkg/archive und devel/old). Das ist ein sicherer Prozess und spart bis zu 500 MB Plattenspeicher ein. Es ist vorteilhaft, das regelmäßig zu tun.
6. Optional: Installieren Sie optionale Sage-Pakete und Datenbanken. Geben Sie in die Eingabe „sage -optional“ ein, um eine Liste zu erhalten, oder laden Sie <http://www.sagemath.org/packages/optional/> und „sage -i <package name>“ zum automatischen Download und zur Installation eines bestimmten Pakets.
7. Optional: Führen Sie die Anweisung „install\_scripts“ innerhalb von Sage aus, um die gp-, singular-, gap-Skripts und weitere Skripts in Ihrem Pfad (PATH) auszuführen. Geben Sie am Sage-Prompt „install\_scripts?“ ein, um Detailinformationen zu erhalten.

Zuletzt: Have fun! Entdecken Sie überraschende Zusammenhänge!

### 4.3 Installation in einer Multiuser-Umgebung

Dieser Abschnitt widmet sich der Frage, wie ein Systemadministrator einen einzige Kopie von Sage für die Multi-User-Nutzung in einem Computer-Netz installieren kann.

#### Systemweit verfügbare Installation

Hier finden Sie eine Zusammenstellung von Mails der „Sage support list“ (speziell solche von Luis Finotti).

1. Entpacken Sie den aktuellen Sage-Tarball (wobei anzunehmen ist, dass es sich um sage- 4.3.4 handelt) z. B. im Verzeichnis /usr/local/ und kompilieren Sie als root. Angenommen Sie arbeiten in einer Root-Shell und der Sage-Tarball befindet sich in Ihrem aktuellen Verzeichnis, dann geben Sie ein:

```
cp sage-2.5.2.tar /usr/local
cd /usr/local
tar xvf sage-2.5.2.tar
cd sage-2.5.2/
```

## 4 Installation aus dem Quellcode

```
make
```

```
(Comment: It's better to build in place. It's a bug if anything goes
wrong when relocating the entire tarball — unfortunately there
is one bug I haven't fixed along these lines, namely the
PARI install hard-codes the location of the "galois data" files.
(Fixes welcome!))
```

Übersetzung: (Kommentar: Es ist besser am Zielort zu kopieren anstatt zuerst zu kompilieren und dann an den Zielort zu verschieben. Es gibt einen Fehler wenn Sie den Tarball reallocieren, weil ein Fehler noch nicht behoben ist, nämlich die PARI-Dateistruktur für den Ort der "galois data", die intern im PARI-Code festgelegt ist. (Korrekturen sind willkommen!))

2. Stellen Sie sicher, dass Sie die Zeile mit den ....“s am Start des Sage-Skripts anpassen. In anderen Worten: Editieren Sie `SAGE_ROOT="....."` zu `SAGE_ROOT="/usr/local/sage-4.3.4"`.
3. Es gibt einige Initialisierungsdateien, die beim ersten Start von Sage erstellt werden müssen. Starten Sie Sage zunächst als Root (oder um genauer zu sein: Führen Sie `test` und den Standardtestcode als Root aus.) Sie können alle Tests mit `Strg+z` und mit der nachfolgenden Eingabe von `kill %1` abbrechen (wobei hier angenommen wurde, dass Sie keine anderen Jobs im Background der Shell laufen haben).
4. Erstellen einer Kopie in `/usr/local/bin`:

```
cp /usr/local/sage-4.3.4/sage /usr/local/bin/
```

Sie erstellen anstatt eines symlinks eine Kopie, weil beim Upgrade mit „`sage -upgrade`“ die Dateien in `/usr/local/sage-4.3.4/sage` überschrieben werden und dabei der Eintrag „`ROOT=...`“ gelöscht wird.

Stellen Sie sicher, dass alle Dateien in `/usr/local/sage-4.3.4` für alle lesbar gesetzt sind:

```
chmod a+rX -R /usr/local/sage-4.3.4
```

### 4.4 Besondere Hinweise

- (Beitrag von Dorian Raymer) Sage-Build schlägt fehl, wenn Sie `bison++` installiert haben. Sie sollten dann `bison++` deinstallieren und `bison` installieren.
- (Beitrag von Peter Jipsen) Wenn Sie einen Fehler in der Art

```
ImportError: /home/jipsen/Desktop/sage-1.3.3.1/local/lib/libpari-gmp.so.2:
cannot restore segment prot after reloc:
Permission denied
```

erhalten, dann hindert Ihre SELinux-Konfiguration Sage am Starten. `launching`. Um das zu korrigieren, können Sie entweder den „`default security context`“ für Sage (??) ändern oder SELinux vollständig deaktivieren. Das erreichen Sie mit dem Eintrag „`SELINUX=disabled`“ in Ihrer Datei `/etc/sysconfig/selinux file`.

- um Sage $\TeX$  für Anwender nutzbar zu machen, lesen Sie die Anweisungen zur Installation in einem „multiuser environment“.

## 5 Sage $\TeX$ mit $\TeX$ bekannt machen

Sage ist weitgehend autark, aber einige Teile bedürfen der Nacharbeit, um optimal zu funktionieren.

Das Paket Sage $\TeX$  erlaubt dem Nutzer Berechnungen und Plots von Sage in  $\TeX$ -Dokumente einzubetten. Sage $\TeX$  ist in Sage bereits installiert, aber um Sage $\TeX$  mit Ihren  $\LaTeX$ -Dokumenten zu nutzen, müssen Sie Sage $\TeX$  Ihrer  $\TeX$ -Installation bekannt machen.

Dazu muss  $\TeX$  Ihre Datei `sagetex.sty` finden, die sich im Verzeichnis `SAGE_ROOT/local/share/texmf/te` befindet, wobei `SAGE_ROOT` der vollständige Pfad zu dem Verzeichnis ist, in dem Sie Sage kompiliert oder installiert haben. Wenn  $\TeX$  die Datei `sagetex.sty` findet, dann arbeitet Sage $\TeX$  korrekt. Es gibt mehrere Wege, das zu realisieren:

- \* Der erste und einfachste Weg ist es, `sagetex.sty` in das gleiche Verzeichnis zu kopieren, in dem sich Ihr  $\LaTeX$ -Dokument befindet. Nachdem das aktuelle Verzeichnis immer durchsucht wird, wenn Sie ein Dokument bearbeiten, funktioniert das immer.

Diese Lösung verursacht aber einige kleine Probleme. Das erste Problem sind mehrere unnötige Kopien von `sagetex.sty` verteilt über mehrere Verzeichnisse.

Das zweite und ernsthaftere Problem ist, dass nach einem Upgrade von Sage mit einer neuen Version von Sage $\TeX$  der Python-Code und der  $\LaTeX$ -Code für Sage $\TeX$  möglicherweise nicht mehr zusammenarbeiten und Fehler verursachen.

- Der zweite Weg ist die Nutzung der Variablen der `TEXINPUTS`-Umgebung. Wenn Sie eine `bash`-Shell nutzen, geht das so:

```
export TEXINPUTS="SAGE_ROOT/local/share/texmf/::"
```

wobei `SAGE_ROOT` wieder der vollständige Pfad des Verzeichnisses Ihrer Sage-Installation ist. Beachten Sie bitte die Zeichen „`///`“ am Ende der Zeile, diese sind sehr wichtig! Danach finden „ $\TeX$  and friends“ die Sage $\TeX$ -Style-Datei. Soll diese Änderung auf Dauer sein, dann können Sie die oben gezeigte Zeile in Ihre Datei `.bashrc` einfügen. Wenn Sie eine andere Shell verwenden, müssen Sie gegebenenfalls die Befehlszeile anpassen so dass die Umgebungsvariable erkannt wird. Ziehen Sie gegebenenfalls das Manual Ihrer Shell zu rate.

Eine Schwäche dieser Methode sind, dass Anwendungen wie `TeXShop`, `Kile`, oder `Emacs/AucTeX` nicht immer die Umgebungsvariable nutzen, sondern  $\LaTeX$  meist außerhalb Ihrer gewöhnlichen Shell ausführen.

Wenn Sie das Verzeichnis Ihrer Sage-Installation verschieben oder eine neue Sage-Version installieren, was gewöhnlich in einem neuen Verzeichnis erfolgt, dann müssen Sie den Eintrag zum neuen Wert von `SAGE_ROOT` korrigieren.



## 5 SageTeX mit TeX bekannt machen

- Der dritte und beste Weg zur Verbindung von TeX und sagetex.sty ist eine Kopie an geeigneter Stelle Ihres Home-Verzeichnisses. Die meisten TeX-Distributionen durchsuchen das Home-Verzeichnis automatisch nach Paketen. Um das passende Verzeichnis zu finden, geben Sie folgendes in der Eingabezeile Ihrer Shell ein:

```
kpsewhich -var-value=TEXMFHOME
```

Danach wird ein Verzeichnis ausgegeben, das ähnlich wie `/home/drake/texmf` oder `/Users/drake/Library/texmf` lautet. Kopieren Sie Ihr tex-Verzeichnis von `SAGE_ROOT/local/share/texmf` in Ihr Home-texmf-Verzeichnis in einer Eingabezeile Ihrer Shell:

```
cp -R SAGE_ROOT/local/share/texmf/tex TEXMFHOME
```

wobei `SAGE_ROOT` wieder wie gewöhnlich den Pfad Ihrer Sage-Installation bezeichnet und `TEXMFHOME` ist die von der Eingabe `kpsewhich` gelieferte Ausgabe (siehe oben).

Wenn Sie Sage upgraden und SageTeX danach nicht mehr funktioniert, wiederholen Sie die beschriebenen Schritte, so dass Sage und TeX wieder synchronisiert werden.

- Zur Installation auf einem Multiuser-System modifizieren Sie die oben beschriebenen Schritte und kopieren Sie `sagetex.sty` in ein geeignetes systemweites TeX-Verzeichnis. Anstelle des Verzeichnisses `TEXMFHOME` ist die Nutzung des Ergebnisses von

```
kpsewhich -var-value=TEXMFLOCAL
```

die bessere Wahl. Die Eingabe liefert ein Verzeichnis, etwa wie `/usr/local/share/texmf`. Kopieren Sie das tex-Verzeichnis wie oben in das Verzeichnis `TEXMFLOCAL`. Wenn Sie die TeX-Datenbasis aktualisieren müssen, können Sie das einfach durch Eingabe von

```
texhash TEXMFLOCAL
```

als root ausführen und `TEXMFLOCAL` anpassen. Danach können alle Nutzer Ihres Systems auf die L<sup>A</sup>TeX-Pakete zugreifen; wenn sie Sage nutzen können, können sie ebenso SageTeX nutzen.

**Warnung** Es ist sehr wichtig, dass die Datei `sagetex.sty`, die L<sup>A</sup>TeX bei der Texterstellung nutzt, mit der Version von SageTeX, die Sage nutzt, zusammenpasst. Wenn Sie Sage aktualisieren, sollten Sie sicherstellen, dass alle älteren, verstreuten Versionen von `sagetex.sty` gelöscht werden.

Wegen dieser Problematik wird dringend empfohlen, die SageTeX Dateien in das Verzeichnis `texmf` in Ihr Home-Verzeichnis (wie oben beschrieben) zu kopieren. Dann ist nach einem Upgrade lediglich ein Verzeichnis zu kopieren und es ist sichergestellt, dass SageTeX richtig arbeitet.

### 5.1 SageTeX-Dokumentation

SageTeX ist nicht grundsätzlicher Teil der Sage-Installation. Dennoch ist es der Erwähnung wert, dass sich die Dokumentation für SageTeX unter `SAGE_ROOT/local/share/texmf/tex/generic/sagetex` befindet.

befindet. In diesem Verzeichnis ist ebenso eine Beispieldatei, siehe `example.tex` und `example.pdf`, das fertige Ergebnis der Bearbeitung dieser Datei mit  $\text{\LaTeX}$  und Sage. Sie können diese Dateien von CTAN herunterladen: <http://www.ctan.org/tex-archive/macros/latex/contrib/sagetex/>.

Sage $\text{\TeX}$  wird zwar „offiziell“ mit Sage ausgeliefert, aber beachten Sie, dass die auf CTAN erhältliche Version nicht unbedingt die neueste Version sein muss.

### 5.2 Sage $\text{\TeX}$ und $\text{\TeX}$ Live

Verwirrend mag sein, dass  $\text{\TeX}$  in der aktuellen Version von  $\text{\TeX}$ Live 2009 die Version Sage $\text{\TeX}$  beinhaltet. Das klingt gut, aber für Sage $\text{\TeX}$  ist es wichtig, dass die Sage-Teile und die  $\text{\LaTeX}$ -Teile synchronisiert werden. Das ist das eigentliche Problem, weil sowohl Sage als auch Sage $\text{\TeX}$  häufig aktualisiert werden,  $\text{\TeX}$ Live hingegen nicht. Aktuell (Januar 2010) liefern viele Linux-Distributionen  $\text{\TeX}$ Live 2007 aus.

Deshalb ist es wichtig und empfehlenswert,  $\text{\LaTeX}$  immer als Teil von Sage $\text{\TeX}$  von Sage zu installieren, so wie das oben beschrieben ist. Die oben beschriebene Anleitung stellt sicher, dass die beiden Teile von Sage $\text{\TeX}$  kompatibel sind, so dass diese problemlos zusammenarbeiten. Eine Nutzung von  $\text{\TeX}$ Live für den  $\text{\LaTeX}$ -Teil von Sage $\text{\TeX}$  wird nicht unterstützt.

## 6 Desktop-Startsymbol

Diese Anleitung zeigt Ihnen, wie Sie unter KDE ein Startsymbol auf dem Desktop zum Starten Ihres Sage-Notebooks erstellen. Die Schritte zu einem Gnome-Desktopsymbol sind wohl ähnlich.

1. Erstellen Sie eine Datei `notebook.sage`, die lediglich die Zeile  
`notebook (openviewer=True)`  
enthält
2. Erstellen Sie in Ihrem Verzeichnis „Desktop“ eine Datei `Sage-notebook.desktop` mit dem folgenden Inhalt:

```
[Desktop Entry]
Comment=
Comment[de]=
Encoding=UTF-8
Exec=/usr/local/bin/sage/home/martin/notebook.sage
GenericName=
GenericName[de]=
Icon=
MimeType=
Name=Sage
Name[de]=Sage
Path=$HOME
StartupNotify=true
Terminal=false
TerminalOptions=
Type=Application
```

## 7 Dokumentation

```
X-DCOP-ServiceType=  
X-KDE-SubstituteUID=false  
X-KDE-Username=
```

Sie müssen in der Zeile Exec= den passenden Pfad zu Ihrer Datei notebook.sage eintragen.

3. Nach Rechtsklick auf das Sage-Notebook Desktopsymbol wählen Sie Eigenschaften, dann Anwendungen, erweiterte Optionen und dort „in Terminal ausführen“. Wenn Sie die Überschrift des Terminalfensters des xwindow anpassen wollen, fügen Sie zu den Terminaloptionen -T "sage notebook" hinzu.

Zum Verlassen des Notebooks drücken Sie zuerst (gleichzeitig) Strg+c im aktiven Terminalfenster, in dem Sage ausgeführt wird, und danach Strg+d um Sage im Terminal zu beenden und schließen Sie zuletzt den Browser oder den Browser-Tab, der das Sage-Notebook Server anzeigte.

Für ein Bild Ihres Startsymbols nutzen Sie den Link zu „Sage art“: <http://wiki.sagemath.org/art>.

## 7 Dokumentation

Die Dokumentation ist Teil von Sage, sie müssen die Dokumentation also nicht unbedingt zusätzlich installieren. Die Dokumentation enthält eine Übung mit Anleitung, ein Handbuch für Entwickler, den „Developers Guide, und diese Installationsanleitung, den Installation Guide in englischer Sprache. Das Tutorial enthält eine Übung mit Anleitung, eine „Guided Tour“ durch Sage und ist ein guter Beginn. Das Referenzhandbuch, „Reference Manual“ beschreibt, was Sage bietet und gibt Beispiele für die Nutzung.

Um eine HTML- oder pdf-Version der Dokumentation zu erstellen, benutzen Sie den Sage-Befehl `-docbuild {document} {format}`. Der Befehl `sage -docbuild reference html` beispielsweise erstellt eine HTML-Version des Referenzhandbuchs; `sage -docbuild -help` zeigt ein Liste aller Optionen.