Einführung in Sage - Einheit 1

Organisatorisches, Was ist Sage?, Streifzug durch Sage

Jochen Schulz

Georg-August Universität Göttingen

Organisatorisches

- Anmeldung über StudIP
  `https://www.studip.uni-goettingen.de/`
  Einführung in Sage (Mathematische Anwendersysteme) (WS 2011/2012)
- Aufgabenblätter und Vorlesungsworksheets sind unter `https://sage.math.uni-goettingen.de` zu erhalten.
- Vorlesungsfolien, Musterlösungen und Zusammenfassungen können aus StudIP heruntergeladen werden.

Organisatorisches

- Anmeldung über StudIP
  `https://www.studip.uni-goettingen.de/`
  Einführung in Sage (Mathematische Anwendersysteme) (WS 2011/2012)
- Aufgabenblätter und Vorlesungsworksheets sind unter `https://sage.math.uni-goettingen.de` zu erhalten.
- Vorlesungsfolien, Musterlösungen und Zusammenfassungen können aus StudIP heruntergeladen werden.

Dozent

Jochen Schulz
NAM, Zimmer 04 (Erdgeschoß)
Telefon: 39-4525 Email: `schulz@math.uni-goettingen.de`
XMPP: `schulz@jabber.num.math.uni-goettingen.de`

Starten des Programms

Vor.: Account im CIP-Pool der Mathematischen Fakultät (MI und NAM): Registrierungs-Formular unter `https://ldap.math.uni-goettingen.de` (Stud.It-Account nötig!)
Intranet/Wiki (`https://wiki.math.uni-goettingen.de`)

- Sage ist in Version 4.7.2 installiert
- login direkt oder per nxclient oder x2goclient auf `login.math.uni-goettingen.de` und `sc1.math.uni-goettingen.de` bis `sc8.math.uni-goettingen.de`
- Nutzen von Sage:
  - Über `https://sage.math.uni-goettingen.de`. Login mit Studentendaten.
  - im Menu unter Education: `sagenotebook` startet (lokale) gui.
  - im Terminal: `sage`

Ablauf der Veranstaltung

- Blockveranstaltung vom 20.2-2.3.2012
- Vorlesung: 9 Uhr bis 11 Uhr
- Übungsbetrieb: 4 Gruppen à je 1h 15min (Besprechung Aufgaben u. Praktikum, Teilnahme freiwillig)
  - 11:00-12:15 (Tutor:)
  - 12:15-13:15 Mittagspause
  - 13:15-14:30 (Tutor:)
  - 14:30-15:45 (Tutor:)
  - 15:45-17:00 (Tutor: J. Vogt)
- Praktikum: von 11:00 bis 19:00 Uhr Computerräume im Keller des MI.
- Übungsbetrieb:
  - 1 Übungszettel/Tag.
  - Klausurzulassung: 3 beliebige markierte Aufgaben/Woche testieren lassen.
  - Alternativ: Projektarbeit durchführen
- Klausur: 9.3.2011; 10:00 - 11:30; Anmeldung über FlexNow.

Inhalt der Vorlesung

1. **Tag** Organisatorisches, Aufbau von Sage, Streifzug durch Sage
2. **Tag** Grundlagen, Symbolisches Rechnen, Gleichungen
3. **Tag** Mengen, Zahlen
4. **Tag** Matrizen, Vektorräume, Funktionen
5. **Tag** Datencontainer, Lineare Abbildungen, Eigenwert und Eigenvektoren
6. **Tag** Folgen, Reihen, Potenzreihen, Vertiefung Schleifen
7. **Tag** Funktionen, Grenzwerte, Funktionenfolgen, Grafiken
8. **Tag** Differentation, Taylorsche Formel, Integration
9. **Tag** Strings, interaktive Grafiken, GeoGebra, Komplexe Beispiele
10. **Tag** Fragestunde

Aufbau

1. **Was ist Sage?**
2. **Streifzug durch Sage**

Aufbau

Computeralgebra-Systeme

Computeralgebra

exakte Berechnungen von mathematischen Objekten

Mathematische Objekte

Natürliche Zahlen, reelle Zahlen, Polynome, Funktionen, Gruppen, Ringe, ...

Numerischen Berechnungen

näherungsweise Berechnung von mathematischen Objekten. Im Computer Gleitpunktdarstellung genannt.

Computeralgebra != Numerische Berechnung

Mathematische Objekte	π, $\sqrt{2}$
Gleitpunktdarstellung (8 Stellen)	3.1415927, 1.4142136

Computeralgebra-Systeme

Allgemein

Sage	Schnittstelle für Mathematik-Software
LiveMath	Maple
Maxima	Free, GPL, von Sage benutzt
Mathematica	einer der Grossen
Maple	einer der Grossen
Matlab	Für große numerische Rechnungen (inkl. Mupad)
Octave	Für große numerische Rechnungen (GPL)
Magma	Spezielle mathematische Rechnungen (z.B. Algebra)
SymPy	Phython-Bibliotheken; als CAS-Verwendbar
SymbolicC++	Bibliotheken zur CA in C++

Überblick:`http://en.wikipedia.org/wiki/Comparison_of_computer_algebra_systems`

Sage

- Ein Open-source (GPL) Mathematik Software System
- Verfügbar seit 24 Februar 2005
- Alternative zu den 4 M's: Magma, Maple, Mathematica, Matlab
- Basiert auf Python
- Objektorientiert
- Besitzt Frontends für viele externe Software
- (Haupt-)Interface im Browser

von Joachim Neubüser (Gründer von GAP):

> *You can read (a) Theorem and its proof [. . .] and then you can use (this) Theorem for the rest of your life free of charge, but for many computer algebra systems license fees have to be paid regularly [. . .]. You press buttons and you get answers in the same way as you get the bright pictures from your television set but you cannot control how they were made in either case.*
>
> *With this situation two of the most basic rules of conduct in mathematics are violated: in mathematics information is passed on free of charge and everything is laid open for checking. Not applying these rules to computer algebra systems that are made for mathematical research [. . .] means moving in a most undesirable direction. Most important: can we expect somebody to believe a result of a program that he is not allowed to see?*

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache
- umfangreiches Hilfesystem

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache
- umfangreiches Hilfesystem
- Viele freie (Unterrichts-)materialien im Internet

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache
- umfangreiches Hilfesystem
- Viele freie (Unterrichts-)materialien im Internet
- Source Code offen und gut dokumentiert (Peer Review)

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache
- umfangreiches Hilfesystem
- Viele freie (Unterrichts-)materialien im Internet
- Source Code offen und gut dokumentiert (Peer Review)

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache
- umfangreiches Hilfesystem
- Viele freie (Unterrichts-)materialien im Internet
- Source Code offen und gut dokumentiert (Peer Review)

Schwächen

- Befehlsumfang nicht so mächtig wie bei Maple, Mathematica oder Matlab

Sage - Stärken und Schwächen

Stärken

- Vereinigung von vielen anderen CAS und Libraries unter einer einheitlichen Oberfläche (Maxima, Pari, GAP, R, Magma, ..., wovon die freien grösstenteils bei Sage enthalten sind)
- Durch Python angebunden an eine mächtige Skriptsprache
- umfangreiches Hilfesystem
- Viele freie (Unterrichts-)materialien im Internet
- Source Code offen und gut dokumentiert (Peer Review)

Schwächen

- Befehlsumfang nicht so mächtig wie bei Maple, Mathematica oder Matlab
- es fehlt eine gute standalone Entwicklungsumgebung (Alternative zum Webinterface: Cantor)

Aufbau

1. Was ist Sage?

2. **Streifzug durch Sage**

Sage als Taschenrechner

Hier einige Beispiele:

```
sage: 3+4*10+12
55
```

```
sage: sin(pi)
0
```

```
sage: float(pi)
3.14159265359
```

```
sage: float(sqrt(2))
1.41421356237
```