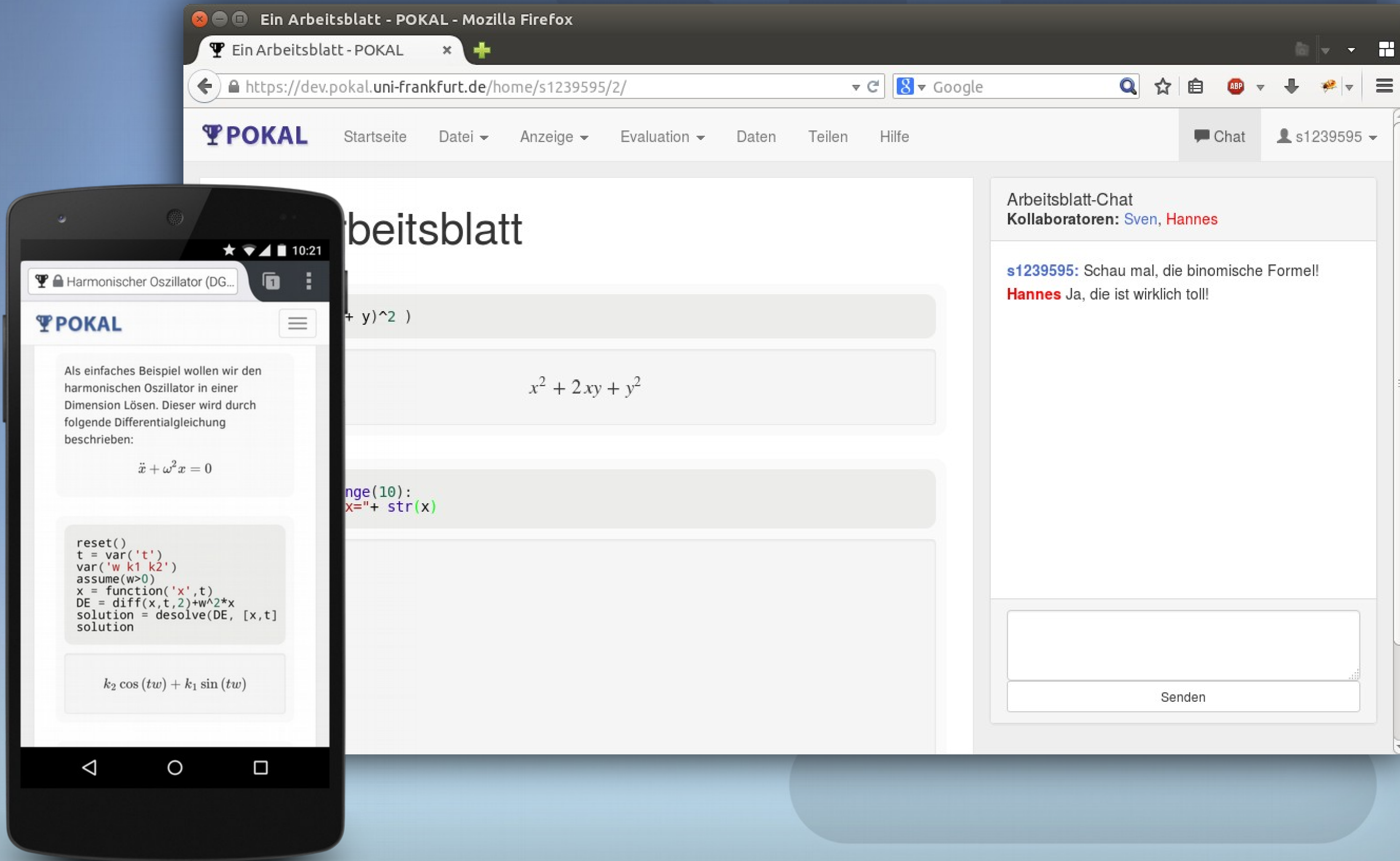
POKAL 2.0: Kollaboratives eLearning neu erfunden

Physik
Onlines
Kollaborative
Arbeits- und
Lernplattform

Carsten Bauer
Sven Köppel

Team PhysikOnline am
Institut für Theoretische Physik

POKAL ist eine Plattform zum Rechnen im Browser



Warum besteht Bedarf?

Kommerzielle Computeralgebra-systeme, zb:

MATLAB
The Language of Technical Computing

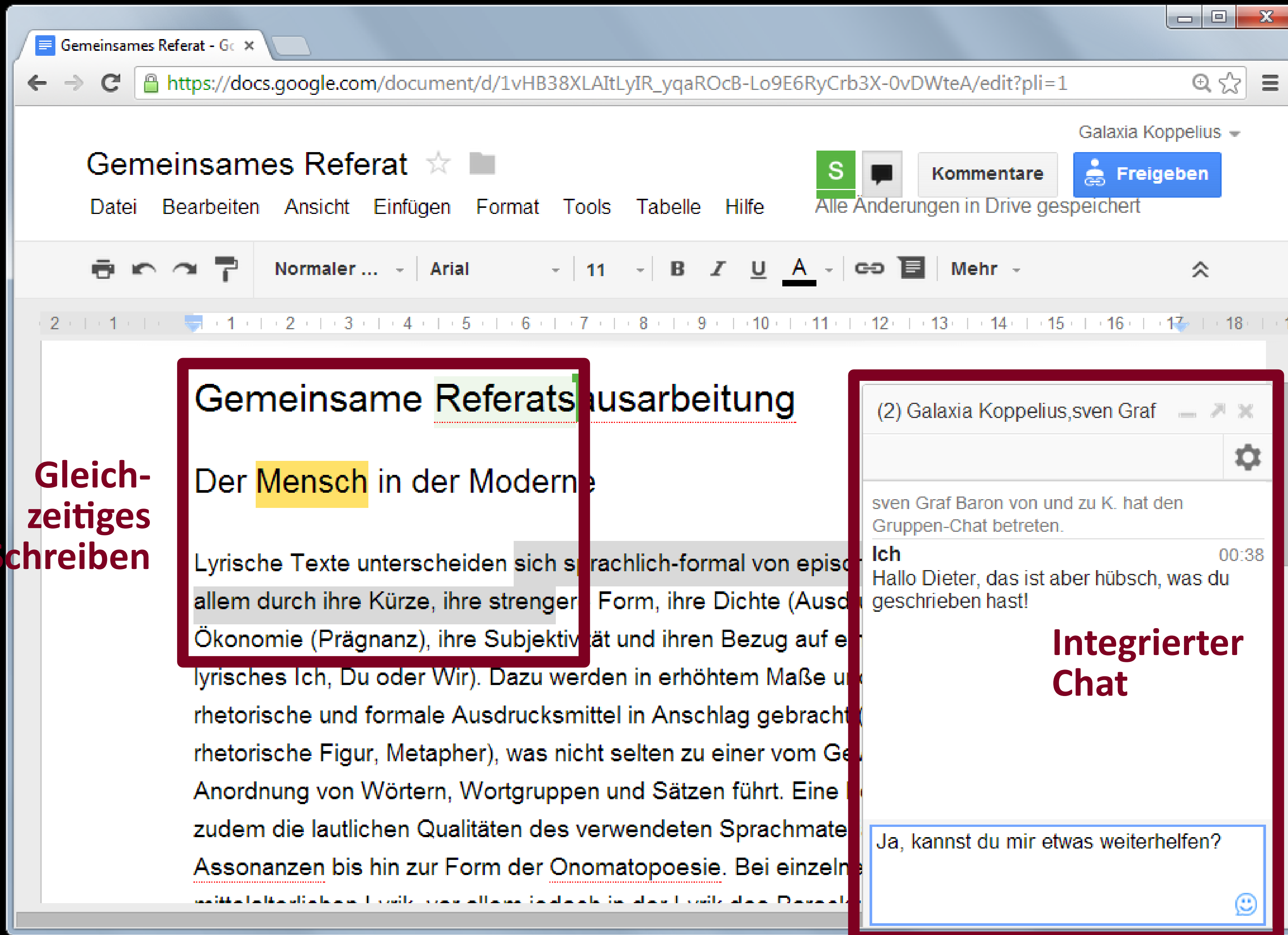
Wolfram Mathematica

- Lizenzkosten! 100.000€/Jahr
- Verfügbarkeit für Studenten: Schwierig
- Vendor-Lockin

Alternativen:
z.B. „Scientific Python“

- Open-Source, kostenlos
- Integrierbar ins Web 2.0

Cloudbasierte Echtzeitkollaborationstools



Google

Das POKAL-Team: am Riedberg

Das POKAL-Team: am Riedberg

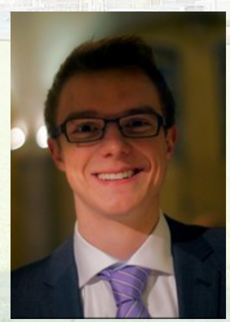
POKAL 1.0

Externer Dienstleister + 2 Hiwis

POKAL 2.0

Thomas

Carsten



Philip

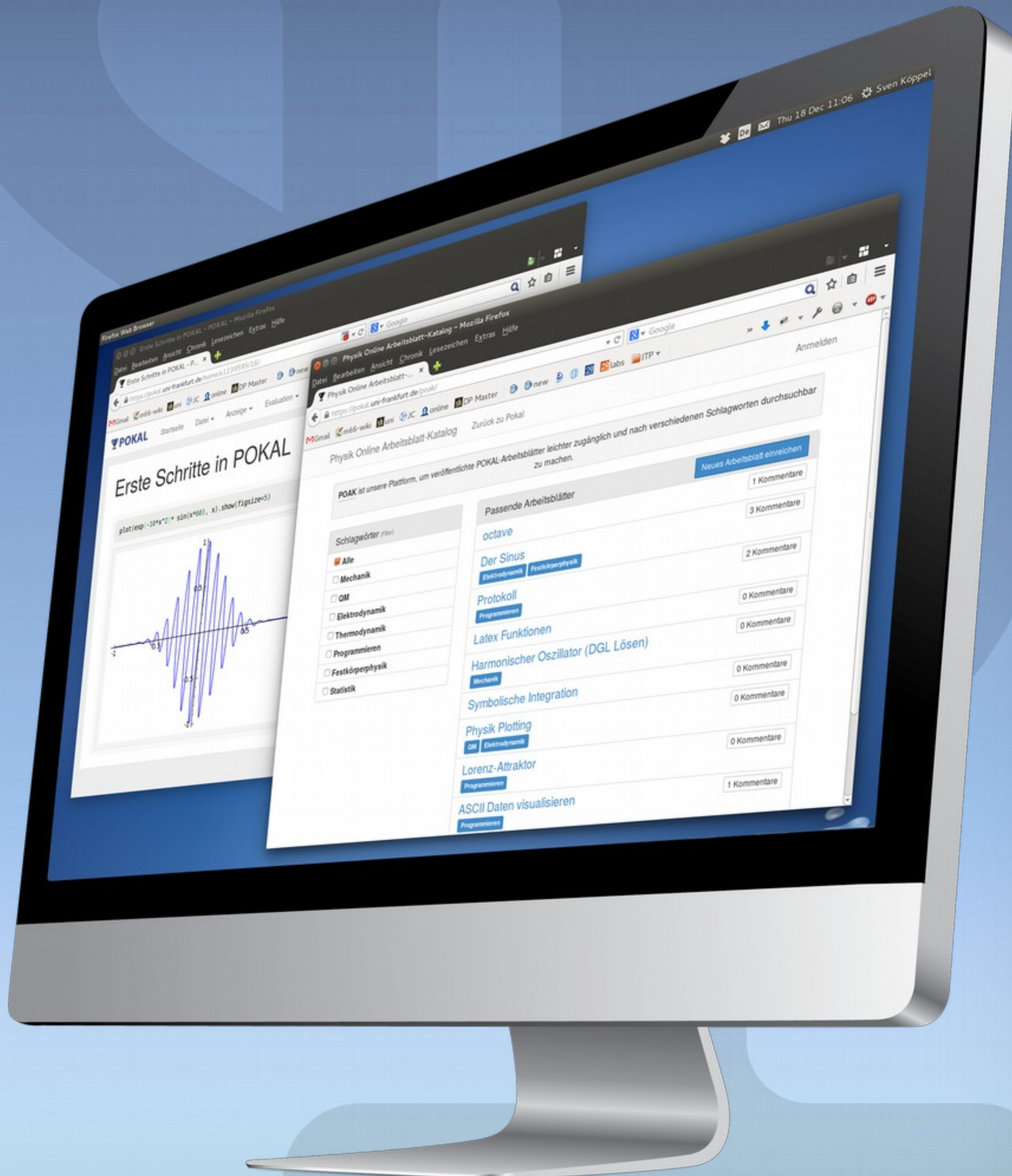
Sven

Die
POKAL-CLOUD

POKAL

EIN PHYSIK-ONLINE-PROJEKT

Einblick



POKAL Startseite PhysikOnline

Willkommen auf POKAL 2.0

Physik Onlines Kollaborative Arbeits- und Lernplattform (POKAL) ist eine brandneue Online-Mathematiksoftware für Studenten und Forscher, die gemeinsam arbeiten, lernen und rechnen wollen. mehr...

Freunde und Förderer

Wir bedanken uns bei unseren Sponsoren und Förderern:

SeLF 2012/2013

POKAL ist ein Pilotprojekt von Physikstudenten und wurde in den Förderrunden 2011/2012 und 2012/2013 im Rahmen des studentischen eLearning-Förderfonds von Studiumdigitale ins Leben gerufen.

ITP

Ohne die Administratoren des ITP und CSC könnten wir die nötige Rechenleistung und Infrastruktur nicht stellen.

SAGE

Das POKAL-Projekt beteiligt sich aktiv an der OpenSource-Mathematik-Software SAGE durch zahlreiche Weiterentwicklungen.

Öffentlicher Arbeitsblatt-Katalog

Benutzer können Arbeitsblätter zu Demonstrationszwecken, für Vorlesungen oder schlicht zum Verbreiten via Link veröffentlichen.

POAK öffnen

Einloggen

HRZ Name

Passwort

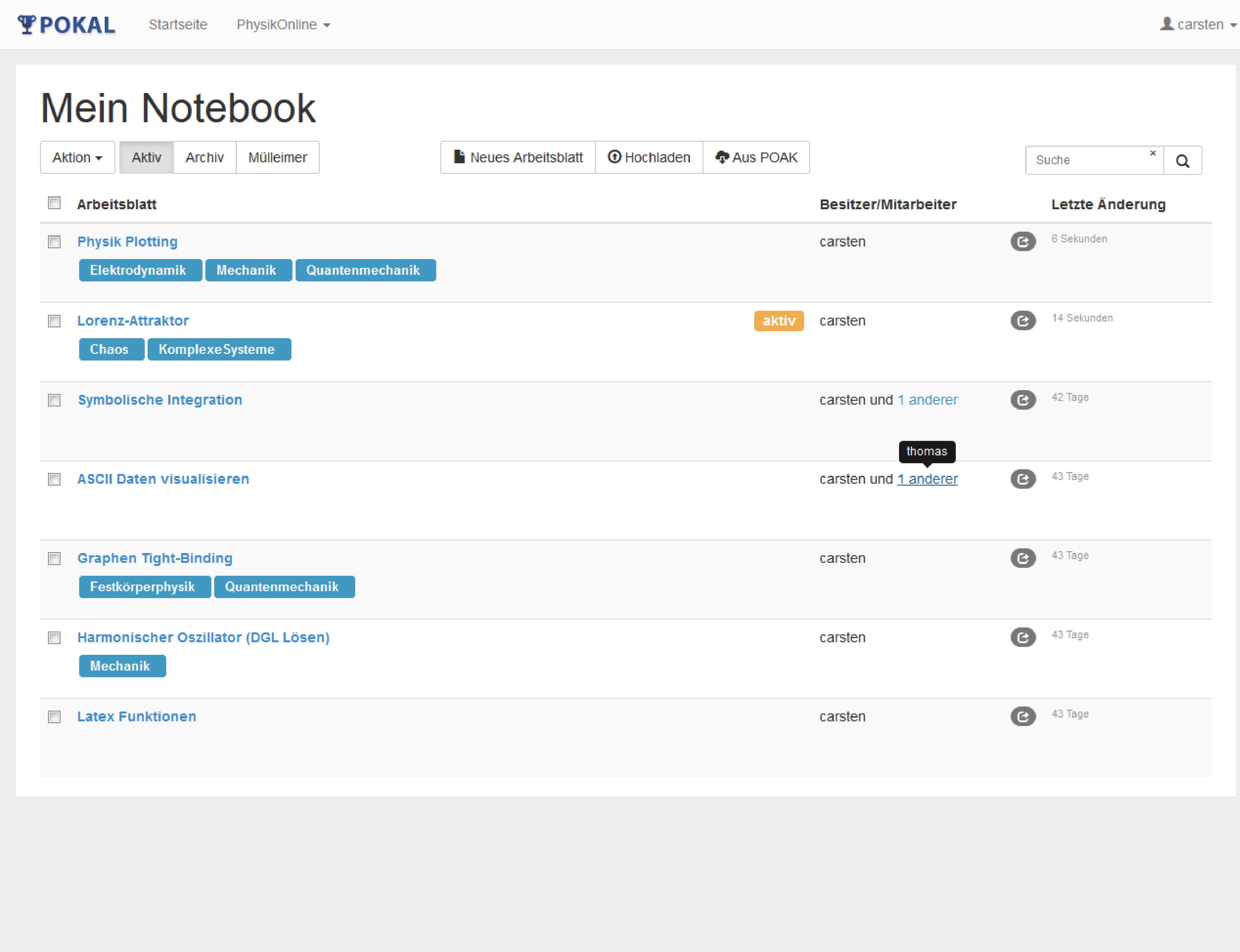
Anmeldedaten merken

Anmelden

Impressum · Datenschutzbestimmungen · Projekt

PhysikOnline ist ein studentisches Projekt der Goethe-Universität

screen 01/10 login



screen 02/10 übersicht

POKAL Startseite Datei Anzeige Evaluation Daten Teilen Hilfe Chat carsten

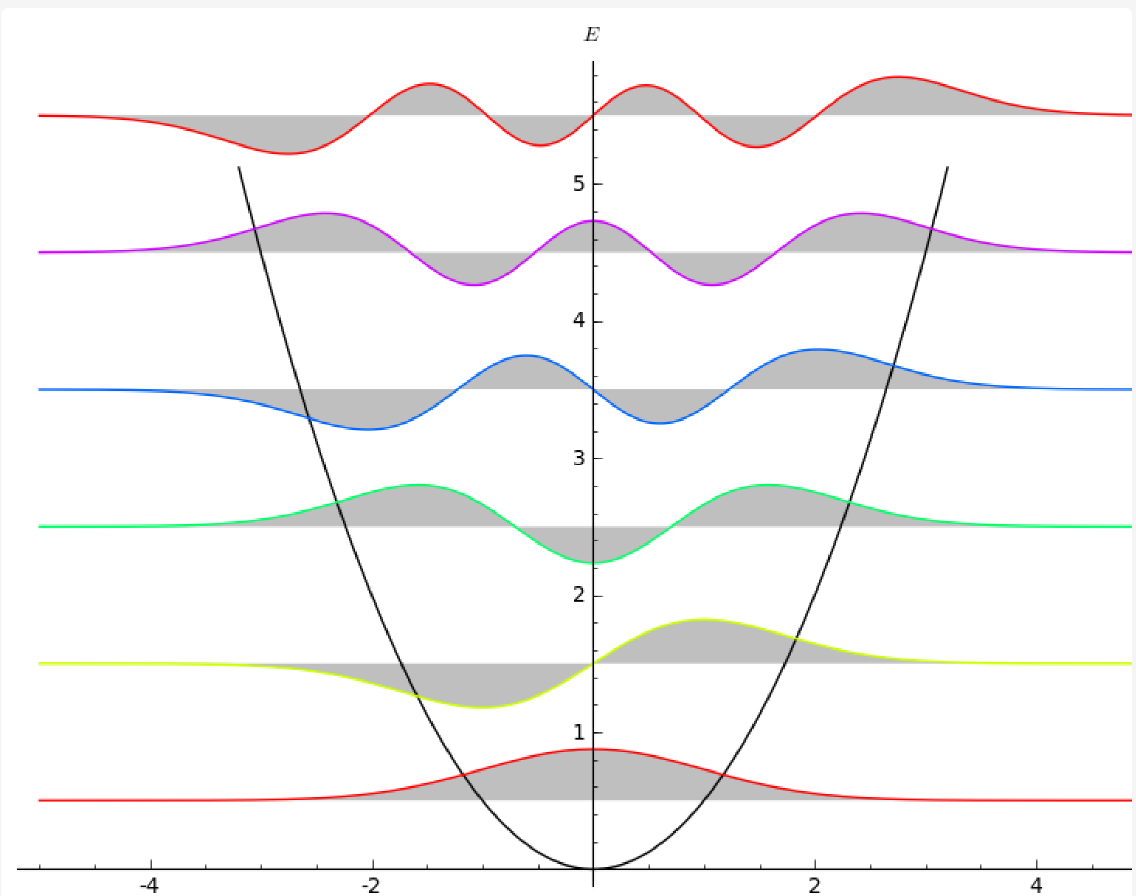
Physik Plotting

```
reset()
def Psi_Harm_Osc(n, x):
 return sqrt(1/(2^n*factorial(n)))/(pi)^(0.25)*e^(-x^2/2)*hermite(n,x)


p = plot(x^2/2, (x, -3.2, 3.2), color='black')

for n in range(6):
 p += plot(Psi_Harm_Osc(n, x)/2 + n + 0.5, (x, -5, 5), color=hue(n/5.0), fill=n + 0.5)

show(p, axes_labels=['$x$','$E$'],figsize=8)
```



Arbeitsblatt-Chat
Kollaboratoren: carsten

carsten: Hier kann diskutiert werden!

carsten: Da auch LaTeX unterstützt wird, können auch Formeln elegant ausgetauscht werden:

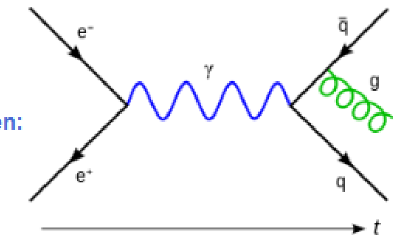
carsten: $\hat{H} = \frac{\hat{p}^2}{2m} + \frac{1}{2} m\omega^2 \hat{x}^2$

carsten:

$$\psi_n(x) = \frac{1}{\sqrt{2^n n!}} \left(\frac{m\omega}{\pi\hbar}\right)^{1/4} e^{-\frac{m\omega x^2}{2\hbar}} H_n\left(\sqrt{\frac{m\omega}{\hbar}}x\right)$$

carsten: Auch diagrammatische Rechnungen lassen sich besprechen:

carsten:

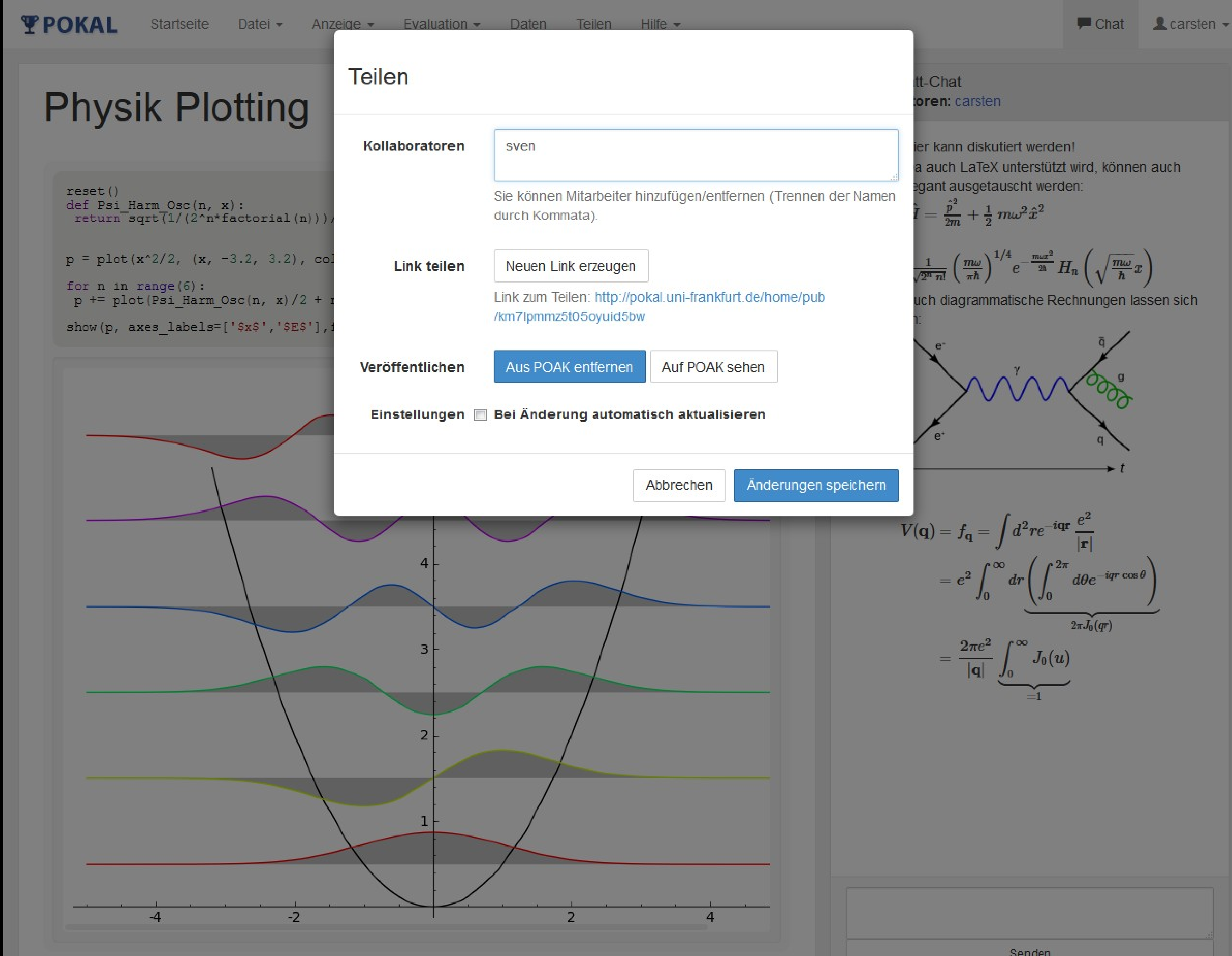


carsten:

$$\begin{aligned} V(\mathbf{q}) = f_{\mathbf{q}} &= \int d^2 r e^{-i\mathbf{q}\mathbf{r}} \frac{e^2}{|\mathbf{r}|} \\ &= e^2 \int_0^\infty dr \underbrace{\left(\int_0^{2\pi} d\theta e^{-iqr\cos\theta}\right)}_{2\pi J_0(qr)} \\ &= \frac{2\pi e^2}{|\mathbf{q}|} \underbrace{\int_0^\infty J_0(u)}_{=1} \end{aligned}$$

Senden

Screen 03/10 chat



Screen 04/10 share

POAK ist unsere Plattform, um veröffentlichte POKAL-Arbeitsblätter leichter zugänglich und nach verschiedenen Schlagworten durchsuchbar zu machen.

Schlagwörter (Filter)

- [x] **Alle**
- [ ] **Mechanik**
- [ ] **QM**
- [ ] **Elektrodynamik**
- [ ] **Thermodynamik**
- [ ] **Programmieren**
- [ ] **Festkörperphysik**
- [ ] **Statistik**

Passende Arbeitsblätter

Neues Arbeitsblatt einreichen

Protokoll
2 Kommentare
Programmieren

octave
1 Kommentare

Der Sinus
3 Kommentare
Elektrodynamik Festkörperphysik

Harmonischer Oszillator (DGL Lösen)
0 Kommentare
Mechanik

Latex Funktionen
0 Kommentare

Symbolische Integration
0 Kommentare

Physik Plotting
0 Kommentare
QM Elektrodynamik

Lorenz-Attraktor
0 Kommentare
Programmieren

Graphen Tight-Binding
0 Kommentare
QM Festkörperphysik

ASCII Daten visualisieren
1 Kommentare
Programmieren

R
0 Kommentare

Harmonischer Oszillator (DGL Lösen)

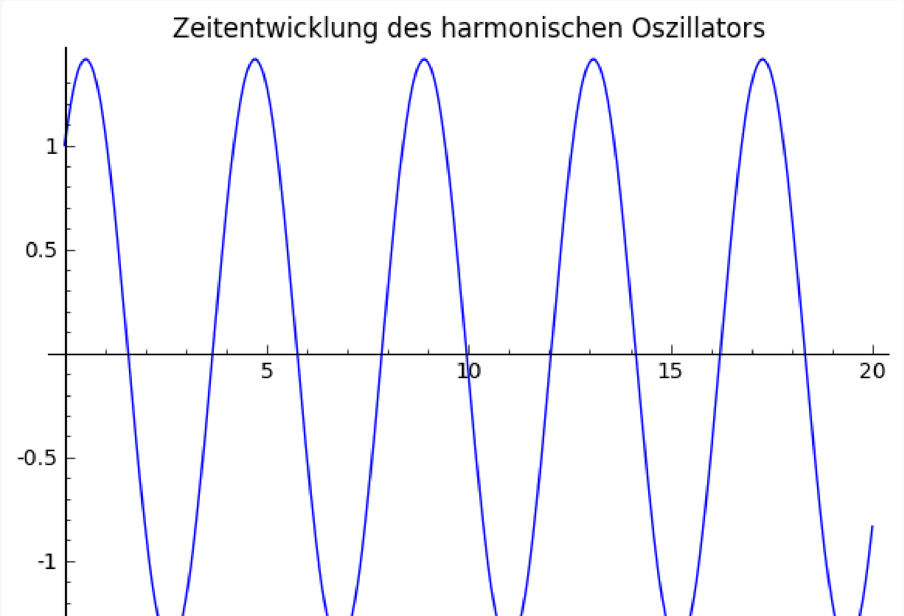
Als einfaches Beispiel wollen wir den harmonischen Oszillator in einer Dimension Lösen. Dieser wird durch folgende Differentialgleichung beschrieben:

$$\ddot{x} + \omega^2 x = 0$$

```
reset()
t = var('t')
var('w k1 k2')
assume(w>0)
x = function('x',t)
DE = diff(x,t,2)+w^2*x
solution = desolve(DE, [x,t])
solution
```

$$k_2 \cos\left(tw\right) + k_1 \sin\left(tw\right)$$

```
pos(t,k1,k2,w) = solution
plot(pos(t,1,1,1.5),(t,0,20),figsize=5,title='Zeitentwicklung des harmonischen Oszillators').show(figsize=6)
```



Lorenz-Attraktor

Der **Lorenz-Attraktor** ist der seltsame Attraktor eines Systems, bestehend aus drei gekoppelten, nichtlinearen gewöhnlichen Differentialgleichungen. Das System ist innerhalb der Chaostheorie ein bekanntes Beispiel für **deterministisches Chaos.** Obwohl die mikroskopische Zukunft des Systems durch die folgenden Differentialgleichungen vollständig determiniert ist, sind praktische Vorhersagen für bestimmte Parameterkonfigurationen unmöglich.

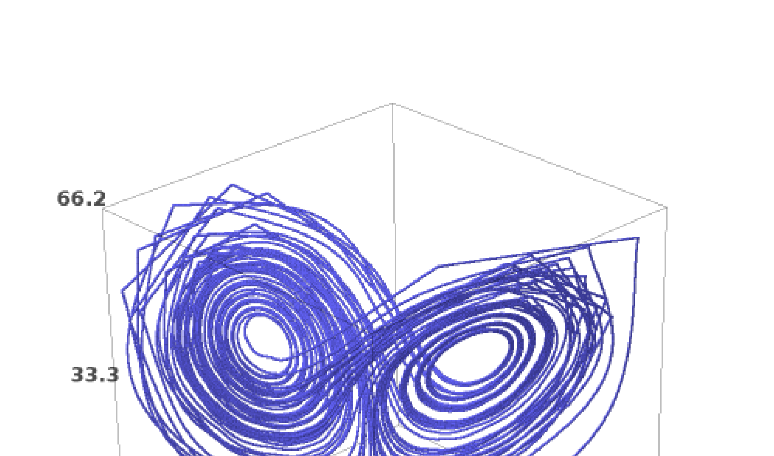
$$\frac{\mathrm{d}x}{\mathrm{d}t} = \sigma(y-x),$$
$$\frac{\mathrm{d}y}{\mathrm{d}t} = x(\rho-z)-y,$$
$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy-\beta z.$$

Wir wollen nun den Lorenz-Attraktor visualisieren. Hierfür lösen wir das System der 3 DGL's numerisch.

```
Integer = int
RealNumber = float
def lorenz(t,y,params):
    return [params[0]*(y[1]-y[0]),y[0]*(params[1]-y[2])- y[1],y[0]*y[1]-params[2]*y[2]]
def lorenz_jac(t,y,params):
    return [ [-params[0],params[0],0],[(params[1]-y[2]),-1,-y[0]],[y[1],y[0],-params[2]],[0,0,0]]
T=ode_solver()
T.algorithm="bsimp" # implicit burlisch-stoer
T.function=lorenz
T.jacobian=lorenz_jac
T.ode_solve(y_0=[.5,.5,.5],t_span=[0,100],params=[10,40.5,3],num_points=10000)
l=[T.solution[i][1] for i in range(len(T.solution))]
line3d(l,thickness=0.3, figsize=5).show()
```

Öffne interaktive Ansicht

Popout



POKAL Startseite Datei Anzeige Evaluation Daten Teilen Hilfe Chat carsten

- Alle Zellen auswerten
- Abbrechen
- Alle Abbrechen
- System ändern
- Alle Ausgaben verbergen
- Alle Ausgaben anzeigen
- Alle Ausgaben löschen
- Weiterrechnen beim Verlassen

Lorenz-Attraktor

Der **Lorenz-Attraktor** ist der seltsame Attraktor eines S … en, nichtlinearen gewöhnlichen Differentialgleichungen. Das System ist innerhalb der Chaostheorie ein bekanntes Beispiel für **deterministis** … ne Zukunft des Systems durch die folgenden Differentialgleichungen vollständig determiniert ist, sind praktische Vorhersagen für bestimmte Paramet …

$$- x),$$

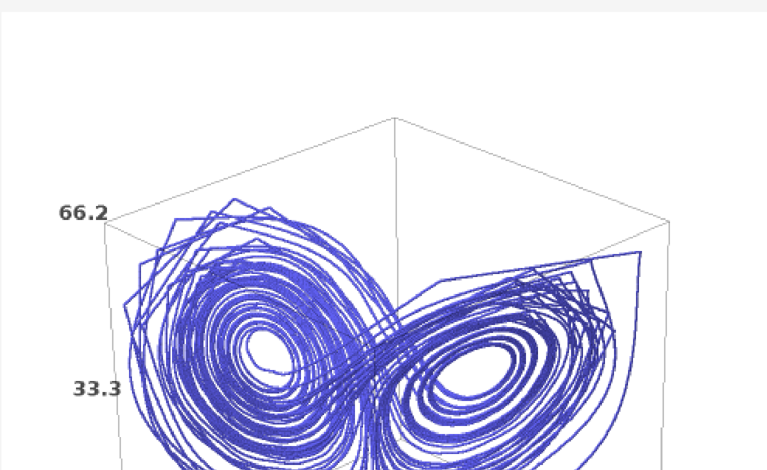
$$- z) - y,$$

$$\frac{\mathrm{d}z}{\mathrm{d}t} = xy - \beta z.$$

Wir wollen nun den Lorenz-Attraktor visualisieren. Hierfür lösen wir das System der 3 DGL's numerisch.

```
Integer = int
RealNumber = float
def lorenz(t,y,params):
    return [params[0]*(y[1]-y[0]),y[0]*(params[1]-y[2])- y[1],y[0]*y[1]-params[2]*y[2]]
def lorenz_jac(t,y,params):
    return [ [-params[0],params[0],0],[(params[1]-y[2]),-1,-y[0]],[y[1],y[0],-params[2]],[0,0,0]]
T=ode_solver()
T.algorithm="bsimp" # implicit burlisch-stoer
T.function=lorenz
T.jacobian=lorenz_jac
T.ode_solve(y_0=[.5,.5,.5],t_span=[0,100],params=[10,40.5,3],num_points=10000)
l=[T.solution[i][1] for i in range(len(T.solution))]
line3d(l,thickness=0.3, figsize=5).show()
```

Öffne interaktive Ansicht Popout



POKAL Startseite Datei Anzeige Evaluation Daten Teilen Hilfe Chat carsten

Graphen Tight-Binding

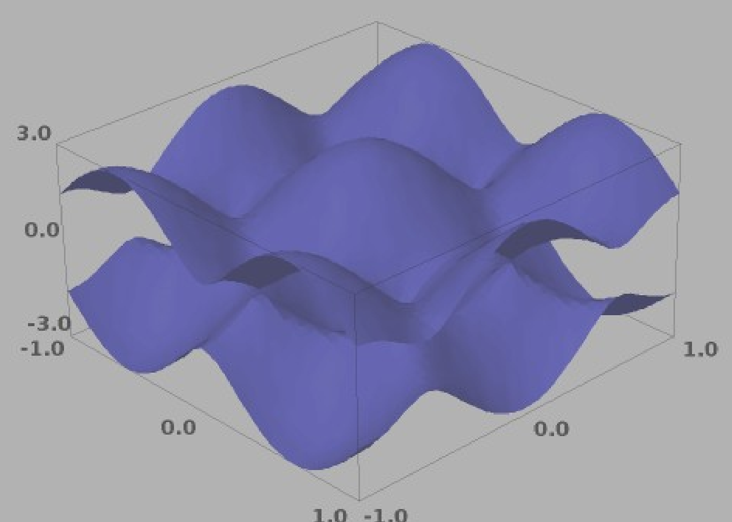
$$E = \pm\sqrt{\gamma_0^2 \left(1 + 4\cos^2 \pi k_y a + 4\cos \pi k_y a \cdot \cos \pi k_x \sqrt{3}a\right)}$$

```
def Ep(kx,ky):
    return sqrt(1+4*cos(pi*ky)^2+4*cos(pi*ky)*cos(pi*kx*sqrt(3)))
%"§)(/")!)!)!"§(!(
def Em(kx,ky):
    return (-1)*sqrt(1+4*cos(pi*ky)^2+4*cos(pi*ky)*cos(pi*kx*sqrt(3)))

p1 = plot3d(Ep,(-1,1),(-1,1))
p2 = plot3d(Em,(-1,1),(-1,1))

(p1+p2).show()
```

Öffne interaktive Ansicht Popout



Problem berichten

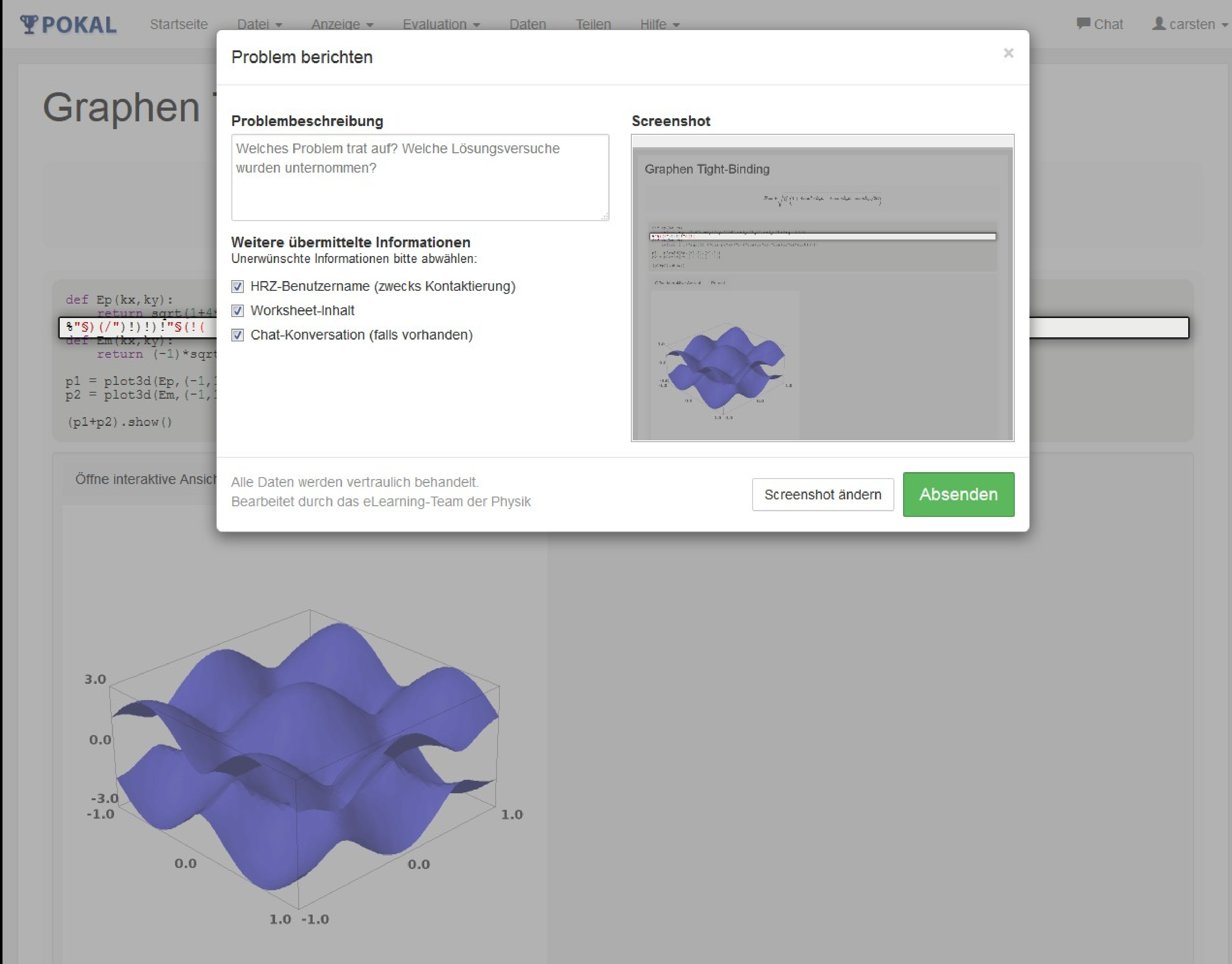
×

Hier besteht die Möglichkeit, dem POKAL-Team einen **Screenshot** samt **Problembeschreibung** zukommen zu lassen. Durch Klicken und Ziehen können Bereiche hervorgehoben oder ausgegraut werden. Falls dieser Dialog im Weg ist, kann er wegbewegt werden.

Hervorheben Problemerelevante Bereiche hervorheben.

Ausblenden Persönliche Informationen verdecken.

Weiter


POKAL
Startseite
Datei
Anzeige
Evaluation
Daten
Teilen
Hilfe
Chat
carsten
Graphen
Problem berichten
Problembeschreibung
Welches Problem trat auf? Welche Lösungsversuche wurden unternommen?
Weitere übermittelte Informationen
Unerwünschte Informationen bitte abwählen:
HRZ-Benutzername (zwecks Kontaktierung)
Worksheet-Inhalt
Chat-Konversation (falls vorhanden)
Screenshot
Graphen Tight-Binding
Alle Daten werden vertraulich behandelt.
Bearbeitet durch das eLearning-Team der Physik
Screenshot ändern
Absenden
def Ep(kx,ky):
return sqrt(1+4
%"§)(/")!)!)!"§(!(
def Em(kx,ky):
return (-1)*sqrt
p1 = plot3d(Ep,(-1,
p2 = plot3d(Em,(-1,
(p1+p2).show()
Öffne interaktive Ansich
3.0
0.0
-3.0
-1.0
1.0
0.0
0.0
1.0 -1.0

POKAL

Kollaboratives eLearning neu erfunden

www.pokal.uni-frankfurt.de
pokal@elearning.physik.uni-frankfurt.de

Carsten Bauer
Sven Köppel

Team PhysikOnline am
Institut für Theoretische Physik