

Schnittstellen Astronomie und Informatik

Markus Pössel, August 2016

Zur Programmiersprache: die meisten Anwendungen, die ich mit Schülern programmiert habe, waren in Python geschrieben. Python hat den Vorteil, dass es in der Astronomie sehr verbreitet ist; deswegen gibt es für diese Programmiersprache zahlreiche Erweiterungen, Module, Bibliotheken, mit denen man z.B. astronomische Originaldaten bearbeiten, auf die Datenbanken von Durchmusterungen zugreifen kann und vieles mehr.

Auch im schulischen Informatikunterricht wird Python erfolgreich eingesetzt; Python als zumindest eine Option für eine schulische Programmiersprache zuzulassen, wäre daher ein wichtiger Vorteil für die astronomischen Bezüge.

Python und Informatikunterricht:

<http://informatik.bildung-rp.de/fortbildung/fortbildungsmaterial/programmieren-mit-python.html>
(Rheinland-Pfalz)

https://dil.inf.tu-dresden.de/fileadmin/dil-web/forschung/e_learning/studentische_arbeiten/ba_birk_web.pdf (Sachsen)

<http://blog.pohlers-web.de/python-die-bessere-programmiersprache-fuer-den-informatikunterricht/> (Plädoyer eines Lehrers aus NRW)

Beispiele für allgemeine Informatik-Kompetenzen, die sich mithilfe astronomischer Beispiel umsetzen lassen:

Umgang mit strukturierten Daten

Beispiel Hertzprung-Russell-Diagramm: Unter ViZier <http://vizier.u-strasbg.fr/viz-bin/VizieR> stehen eine Vielzahl astronomischer Kataloge zur Verfügung. Eine spanische Schülerin in unserem diesjährigen Sommerpraktikum erstellt aus dem dort als CSV-Dokument verfügbaren Hipparcos-Katalog http://vizier.u-strasbg.fr/viz-bin/VizieR-3?-source=I/239/hip_main&-out.max=50&-out.form=HTML%20Table&-out.add=r&-out.add=RAJ,DEJ&-sort=r&-oc.form=sexa derzeit ein Hertzprung-Russell-Diagramm. Dazu muss sie astronomisch verstehen

- Das Helligkeitsklassen-System verstehen
- Absolute und scheinbare Helligkeiten mit Entfernung verknüpfen
- Ein qualitatives Verständnis von Farbe (Magnituden-Differenz) und Temperatur haben
- Fortgeschritten: eine Näherungsgleichung für Hauptreihensterne ableiten

und informatikbezogen

- Eine Textdatei (ähnlich CSV) öffnen und in ein Array einlesen
- Dieses Array elementweise manipulieren (Schleife über die Einzelwerte; aus den Einzelwerten für Parallaxen die Entfernungen berechnen; aus den Einzelwerten für gemessene Helligkeiten und Entfernungen die absoluten Helligkeiten)
- Die Ergebnisse zweidimensional plotten (was mit matplotlib sehr einfach geht)

Umgang mit Datenbanken

Beispielsweise für den Sloan Digital Sky Survey (SDSS) stehen Daten für Millionen von Sternen, Galaxien und Quasaren zur Verfügung: Helligkeiten, Rotverschiebungen, sogar Spektren. Mit dem SciServer <http://www.sciserver.org/> bietet der SDSS sogar eine frei zugängliche Umgebung, in der Python-Programme ausgeführt werden können (Jupyter-Notebooks), die dann direkt und sehr schnell auf die SDSS-Datenbank zugreifen können.

Beispiel Helligkeit von Standardkerzen: Über das invers-quadratische Gesetz für die Helligkeit und die (angenommen konstante) absolute Helligkeit bekommt man einen direkten Zusammenhang zwischen Entfernung und scheinbarer Helligkeit m . Die Entfernung holen wir uns bei entfernten Galaxien über die Rotverschiebung z . Dann vergleichen wir die Kurve $m(z)$, die sich so ableiten lässt, mit einem Plot der SDSS-Galaxien in einem z - m -Koordinatensystem. Die untere Einhüllende des Scatterplots entspricht tatsächlich in guter Näherung unserer Standardkerzen-Kurve. Schluss: Die absolut hellsten Galaxien sind Standardkerzen, also immer ungefähr gleich hell!

Astronomisch muss man dafür verstehen:

- Helligkeits-Abstandsgesetz
- Astronomisches Helligkeitssystem
- Hubble-Beziehung

Und aus der Informatik kommt dazu:

- Verbindung mit einer Datenbank
- Abfrage einer Datenbank (beim SDSS: vereinfachte Version von SQL)
- Umgang mit Array-Daten, Manipulation von Array-Elementen
- Plotten von Punkten und Kurven

Eine ähnliche Auswertung zum Nachweis der Linearität der Rotverschiebungs-Helligkeitsbeziehung ist hier ausgearbeitet: <http://www.wissenschaft-schulen.de/alias/material/online-auf-den-spuren-von-hubble-und-wirtz/1051374>

Simulationen: Grundlagen

Simulationen, also die Modellierung realer Systeme im Computer samt Berechnung von deren Eigenschaften, insbesondere der zeitlichen Entwicklung, ist eine zentrale Anwendung von Computern – auch in der Astronomie.

Programmiertechnisch lassen sich einfache Simulationen ohne große konzeptuelle Schwierigkeiten umsetzen. Benötigt werden lediglich Schleifen (z.B. für die aufeinanderfolgenden Zeitschritte), die Möglichkeit Variablen zu modifizieren (einfache Rechenoperationen wie „neue Geschwindigkeit gleich alte Geschwindigkeit plus Zeitschritt mal Beschleunigung“), ggf. Abfragen besonderer Bedingungen und das numerische/analytische Lösen der entsprechenden Gleichungen (z.B.: sind sich zwei Teilchen, die Kugeln darstellen, so nahe gekommen, dass sie sich anstoßen?) sowie für die Anschaulichkeit Möglichkeiten, das Ergebnis grafisch darzustellen.

Beliebtes Beispiel aus dem schulischen Informatikunterricht: ein Billard-Spiel, bei dem die Kugeln mit Anfangsgeschwindigkeit loslaufen und an den Grenzen des Billardtisches oder bei Kollisionen miteinander sprunghaft ihre Richtung ändern.

Astronomisches Beispiel 1: Raytracing. Das Verhalten von Licht in Fernrohren lässt sich ganz analog zum herkömmlichen Billard analysieren. Außerhalb der optischen Elemente läuft das Licht geradlinig. Wann und wo das Licht auf eine Grenzfläche (Linsenoberfläche oder Spiegeloberfläche) trifft, lässt sich auf ähnliche Weise berechnen wie für die Kollision mit Billardkugeln/Tischgrenze (mathematisch: Schnittpunkte einfacher Funktionen). Das Licht ändert seine Richtung entsprechend den Reflexions-/Brechungsgesetzen.

So lassen sich die grundlegenden Abbildungsgesetze für die verschiedenen Typen astronomischer Teleskope (insbes. auch Vergrößerung, Lichtsammelvermögen) direkt simulieren – in Verbindung von Informatik, Optik und Astronomie.

Astronomisches Beispiel 2: Bedeckungen. Ohne in die dynamischen Details gehen zu müssen, lassen sich Lichtkurven simulieren, wie sie bei Doppelsternsystemen entstehen oder bei Sternen, die von Planeten umkreist werden (Exoplaneten-Transits). Das ist mit jeder Software möglich, die es erlaubt, geometrische Formen (zweidimensional als Kreise oder dreidimensional als Kugeln) zu verschieben, daraus Bilder zu erstellen und deren Gesamthelligkeiten auszuwerten (siehe meinen WIS-Beitrag hier: <http://www.wissenschaft-schulen.de/alias/material/schattenspiel-mit-fremden-welten-exoplaneten-lichtkurven-einfach-simulieren/1128724>)

Die Simulationen - oder alternativ geometrische Formeln für Lichtkurven, die allerdings Integration voraussetzen – können mit echten Lichtkurven verglichen werden, wie sie beispielsweise die NASA in Form der Messungen des Kepler-Satellitenteleskops zur Verfügung stellt:

http://archive.stsci.edu/kepler/data_search/search.php

Simulationen: numerische Fehler und Zuverlässigkeit

Bei allen Simulationen stellt sich die Frage nach numerischen Fehlern und der Zuverlässigkeit: Was von dem, was mir der Computer liefert, entspricht den

physischen (oder anderweitig realen) Gegebenheiten, was ist Artefakt der Simulation?

Hier bieten sich dynamische Simulationen aus der Physik an: Teilchen, die mithilfe von Kräften aufeinander wirken; aus den Kräften ergibt sich, wie sich die Beschleunigungen der Teilchen mit der Zeit ändern, aus den Beschleunigungen wiederum die Geschwindigkeitsänderungen, aus jenen die Ortsänderungen.

Schönes Beispiel für die potenziellen Probleme numerischer Simulationen ist die Simulation des harmonischen Oszillators (einzelnes Teilchen mit Hook'scher Rückstellkraft zum Nullpunkt): mit dem herkömmlichen Eulerverfahren schaukeln sich die Ungenauigkeiten rasch auf; das Leapfrog-Verfahren bringt bereits eine deutliche Verbesserung. Der Weg in Richtung Runge-Kutta kann SuS zumindest über eine polynomiale Näherung (anstatt einer richtigen Taylor-Entwicklung) nahegebracht werden.

Vielfältige **Anwendungsbeispiele** liefert die Himmelsmechanik:

- Umlaufbahnen von Planeten, Asteroiden, Kometen (dabei z.B. Vergleich der numerisch errechneten Bewegung mit den Kepler'schen Gesetzen, insbesondere dem 2. und 3.)
- Resonanzphänomene in Mehrkörpersituationen, z.B.: Sonne als Zentralkörper, Jupiter als zweite Gravitationsquelle, weitere Punktteilchen als Asteroiden: Kirkwood-Lücke, Trojaner und ähnliche Resonanzen.
- Raumfahrtsimulationen (einer Reihe von SuS vermutlich schon über das Computerspiel Kerbal Space Program) bekannt: Wie bekommt man einen Satelliten in den Orbit? Warum liegen Abschussstellen nahe des Äquators? Wie bekommt man Astronauten zum Mond oder zum Mars? (zu letzterem Thema z.B. der WIS-Beitrag „Der Weg zum Mars“, <http://www.wissenschaft-schulen.de/alias/material/der-weg-zum-mars/1128723>), Flug/Rücksturz der neuen Orion-Kapsel (<http://www.wissenschaft-schulen.de/alias/material/orion-kontrollierter-ruecksturz-zur-erde/1183912>), Landung auf dem Kometenkern von Rosetta (<http://www.wissenschaft-schulen.de/alias/material/landung-auf-einem-kometenkern-etwas-schulphysik/1285849>)
- Spezialfall: Auch Fluchtgeschwindigkeiten lassen sich auf diese Weise simulieren, mit der Möglichkeit einer direkten Verknüpfung zur Mechanik und zu spannenden astronomischen Themen wie Schwarzen Löchern
- Relativistische Simulationen: In pseudo-Newton'scher Näherung bewegt sich auch Licht wie andere Teilchen im Gravitationsfeld, nur eben mit asymptotischer Anfangsgeschwindigkeit c . So lässt sich mit einer einfachen dynamischen Simulation beispielsweise die Lichtablenkung am Sonnenrand (Eddington-Expedition 1919) simulieren.

Fitten von Daten und Funktionszusammenhängen

Das Anpassen von Funktionalzusammenhängen (Funktionen mit freien Parametern) an Messdaten ist eine zentrale Kompetenz nicht nur in der Physik, sondern in allen quantitativen Wissenschaften. Hier hat die Informatik den großen Vorteil, dass die einfachsten Anpassungs-Verfahren eine sehr einfache algorithmische Umsetzung haben. Die Methode der kleinsten Quadrate beispielsweise reduziert sich darauf, für eine vorgegebene Menge von Parameter-Werten (bzw. allgemeiner Wertepaaren oder -tupeln) eine Kenngröße (Summe der quadratischen Abweichungen) zu bestimmen und am Ende zu vergleichen, für welchen Parameterwert die Kenngröße am kleinsten ist.

Das lässt sich als Programm realisieren, sobald die SuS Array-Strukturen, for-Schleifen und if-Bedingungen mit Vergleichen beherrschen – weit niedrigere Voraussetzungen, als für ein rein mathematisches Verständnis der Methode der kleinsten Quadrate notwendig wären!

Astronomisches Beispiel 1: Hubble-Beziehung und beschleunigtes Universum. An (vorher ausgewählte) Daten für die scheinbaren Helligkeiten (wahlweise gleich: Entfernungen) und Rotverschiebungen von Supernovae vom Typ Ia wird eine quadratische Kurve angepasst, um Hubble-Konstante und Beschleunigungsparameter zu bestimmen – und damit immerhin wichtige Erkenntnisse des Physik-Nobelpreises 2011 nachzuvollziehen!

Astronomisches Beispiel 2: Spektren fitten – drei Teilnehmerinnen unseres diesjährigen Sommerpraktikums fitten zur Bestimmung der Temperatur Planck-Kurven an die aus dem SDSS entnommenen Spektren (Fits-Dateien, in Python geöffnet) von Sternen.

Digitale Bildbearbeitung und -analyse

Digitale Bildbearbeitung ist ein Aspekt der Informatik, mit dem so gut wie alle SuS bereits in Berührung gekommen sind – vom Zuschneiden eigener Fotos bis zu Instagram-Filtern, Augmented Reality bei Tablets und Smartphones und den aufwändigen digitalen Spezialeffekten von Film- und Fernsehproduktionen.

In der Amateur- und Schulastronomie ist die digitale Bildbearbeitung seit Jahren weit verbreitet - dank erschwinglicher Digitalkameras.

Für den Schulunterricht stehen daher zahlreiche Möglichkeiten zur Erstellung digitaler Bilder zur Verfügung: mit eigener Digitalkamera und einfacher Linse, mit eigener Digitalkamera oder Webkamera am Kleinteleskop, oder über Beobachtungen mit für Schulprojekte zugänglichen größeren Teleskopen wie ROTAT (<http://www.stiftung-astronomie.de/de/ueber-rotat>), MONET (<https://monet.uni-goettingen.de/>), Faulkes Telescopes (<http://www.faulkes-telescope.com/education>) oder LCOGT (<https://lcogt.net/education/>).

Die einfachste Auswertung digitaler Bilder nutzt allgemeine Bildbearbeitungssoftware wie Photoshop und GIMP (<https://www.gimp.org/>) oder wissenschaftliche / astronomische Software, die für den Amateur- oder

Bildungsbereich geeignet ist, wie ImageJ (<https://imagej.nih.gov/ij/>) oder AstroImageJ (<http://www.astro.louisville.edu/software/astroimagej/>)

Allgemeine, Mittelstufentaugliche Aktivitäten zum Kennenlernen von und Umgang mit astronomischen digitalen Bildern bietet <http://www.wissenschaft-schulen.de/alias/material/sendem-und-empfangen-digital-verschlueselter-bilder/1070006>

Astronomisches Beispiel 1: Auswertung der Bahn des Meteoriten von Tscheljabinsk mithilfe des Schattenwurfs von Straßenlaternen auf einem Video <http://www.wissenschaft-schulen.de/alias/material/meteoroid-von-tscheljabinsk-bahnberechnung-im-unterricht/1285850>

Astronomisches Beispiel 2: 3D-Bilder zur Marserkundung (inklusive der Erstellung eigener 3D-Bilder, die mit einer Rot-Blau-Brille dreidimensional gesehen werden können): <http://www.wissenschaft-schulen.de/alias/material/mars-und-erde-im-vergleich/1063534>

Astronomisches Beispiel 3: Auswertung von Sonnenfleckebildern von NASA-Sonden: Rekonstruktion der Positionen und der Rotationsperiode der Sonne mithilfe von Geogebra, aber auch mit anderer Software möglich <http://www.wissenschaft-schulen.de/alias/material/auswertung-von-sonnenfleckebildern/1116795>

Astronomisches Beispiel 4: Mit geeigneter Software, z.B. Python mit der Bibliothek astropy, kann man auch direkt auf astronomische Bilder im FITS-Format zugreifen. So lassen sich Sternhelligkeiten (Aperturphotometrie, in der Regel nicht kalibriert) und Sternpositionen (Fitten von Gauß-Profil an ein Sternscheibchen) selbst bestimmen. Mögliche Anwendungen: HR-Diagramm für einen Sternhaufen, Positionsbestimmung von Planeten / Kleinplaneten, Selbst angefertigte Farbbilder aus Bildern in drei Filtern (Verschiebung der Filterbilder entsprechend von Referenz-Sternpositionen).

Astronomisches Beispiel 5: Abstandsgesetz: Mithilfe einer Standard-Lichtquelle (LED), die mit einer Digitalkamera aus zunehmend größerer Entfernung gemessen wird (Maßband), kann das invers-quadratische Abstandsgesetz für die Helligkeit gemessen werden. Die Helligkeit wird dabei aus der aufsummierten Pixelhelligkeit (im wesentlichen: Apertur-Fotometrie) des Lichtquellenabbilds auf den digitalen Bildern bestimmt.

Astronomisches Beispiel 6: Alltags-Standardkerzen: Für gleich helle Straßenlaternen kann analog zur Standardkerzenmethode die Entfernung bestimmt werden. Dabei werden auf einem mit Digitalkamera aufgenommenen Bild die relativen scheinbaren Helligkeiten bestimmt und daraus die relativen Entfernungen erschlossen. Das Ergebnis kann z.B. mithilfe von Google Maps überprüft werden.

Astronomisches Beispiel 7: Parallaxe. Mithilfe einer Digitalkamera werden zwei Bilder eines nahen Objekts (z.B. Ball auf Stativ) und eines fernen

Hintergrundes gemacht. Zwischen dem ersten und dem zweiten Bild wird die Kamera seitlich verschoben. Aus der Verschiebung zwischen Objekt und Hintergrund zwischen dem ersten und zweiten Bild kann mittels Parallaxeneffekt auf die Entfernung des Objekts geschlossen werden.

Astronomisches Beispiel 8: Highspeed-Kameras. Einige Consumer-Digitalkameras können (mit begrenzter Auflösung!) bis zu 1000 Bilder pro Sekunde schießen. So lassen sich beispielsweise simulierte Asteroideneinschläge – Stein fällt auf Mehluntergrund – untersuchen und zeigen, wie Ejekta bei Einschlagskratern zustande kommen.

Astronomisches Beispiel 9: Lichtkurven. Anhand eigener oder vorhandener digitaler Aufnahmen ein und desselben Objekts können die Lichtkurven variabler Sterne, im Extremfall sogar von Exoplaneten (Transits) bestimmt werden.

Signalverarbeitung und Kommunikation

Ein einfaches Projekt, Satellitendaten mithilfe eines USB-Adapters direkt im Computer zu decodieren und auszuwerten, wird hier vorgestellt :
<http://www.wissenschaft-schulen.de/alias/material/nach-hause-telefonieren-kommunikation-zwischen-raumsonden-und-der-bodenstation/1128719>