



Algorithmisch denken im Mathematikunterricht

–

Programmieren mit Python

Seminar im SS 2021
im Rahmen von MINT-Lehramt PLUS

- 1 Organisatorisches
- 2 Einstieg in Python
- 3 Mathematische Verfahren erschließen, Algorithmen und Programme entwickeln und nutzen
- 4 Fachdidaktische Impulse

1 Organisatorisches

1.1 Steckbrief

Inhalt	<ul style="list-style-type: none">▪ Algorithmische Verfahren der Sekundarstufenmathematik kennenlernen bzw. vertiefen und an einem Rechner umsetzen▪ Mit der Sprache Python programmieren▪ Verfügbare Lernangebote zu Python didaktisch reflektieren▪ Über algorithmisches Denken im Mathematikunterricht reflektieren▪ Möglichkeiten der differenzierten Förderung von Schülern diskutieren
Zielgruppe	Mathematikstudierende in MINT-Lehramt PLUS
Betreuer	Prof. Dr. Volker Ulm, Universität Bayreuth, Lehrstuhl für Mathematik und ihre Didaktik
Lernform	<ul style="list-style-type: none">▪ Eigenständiges und kooperatives Erarbeiten von Grundlagen des Programmierens mit Python, Erarbeiten algorithmischer Verfahren der Sekundarstufenmathematik, Umsetzen dieser Verfahren an einem Computer▪ Informeller Austausch in der Gruppe aller Teilnehmer, gemeinsames Diskutieren von mathematischen, informatische und fachdidaktischen Fragen▪ Austausch mit digitalen Medien (z. B. Zoom, E-Mail) und ggf. Treffen in Kleingruppen
Modul	Passend für die Module C1, C2, D1, D2 in MINT-Lehramt PLUS (unbenotet), nach besonderer Absprache auch für die Module B1, B2 (benotet)
Zeitraum	April bis August 2021

1.2 Intention

Das Seminar verfolgt mehrere miteinander verwobene Ziele:

- *Mathematik:* Sie lernen Inhalte der Sekundarstufenmathematik kennen bzw. vertiefen diesen, bei denen die Berechnung von Zahlen im Fokus steht (z. B. ggT, kgV, Primfaktorzerlegung, Wurzeln, Kreiszahl π , Eulersche Zahl e , Nullstellen von Funktionen, numerische Integration).
- *Programmieren:* Sie lernen die Programmiersprache Python so weit kennen, dass Sie die erarbeiteten mathematischen Verfahren jeweils in einem Programm umsetzen können, das die Berechnungen am Computer ausführt.
- *Didaktik:* Sie reflektieren das Potential der erarbeiteten mathematischen Inhalte für die Gestaltung von differenzierendem Mathematikunterricht. Sie reflektieren frei verfügbare Lernangebote zum Einstieg in das Programmieren mit Python im Hinblick auf Ihr eigenes Lernen und auf möglichen Nutzen für Schüler.

Die sind insbesondere Grundlagen, um als Lehrkraft Schülerinnen und Schülern Impulse zum Erarbeiten der jeweiligen mathematischen Inhalte und zum Umsetzen von Algorithmen mit Programmen zu geben.

1.3 Ablauf

Das Seminar beginnt im Rahmen der Auftaktveranstaltung zum Sommersemester für MINT-Lehramt PLUS am 07./08. April 2021 mit zwei jeweils einstündigen Video-Treffen:

Treffen 1	Mittwoch, 07. April 2021	13:00 – 14:00 Uhr
Treffen 2	Donnerstag, 08. April 2021	13:00 – 14:00 Uhr

Im Abstand von zwei bis drei Wochen finden fünf weitere Treffen im Lauf des Sommersemesters statt:

Treffen 3	Freitag, 30. April 2021	16:00 – 17:30 Uhr
Treffen 4	Freitag, 21. Mai 2021	16:00 – 17:30 Uhr
Treffen 5	Freitag, 11. Juni 2021	16:00 – 17:30 Uhr
Treffen 6	Freitag, 25. Juni 2021	16:00 – 17:30 Uhr
Treffen 7	Freitag, 09. Juli 2021	16:00 – 17:30 Uhr

Den Zoom-Link erhalten Sie mit der Anmeldung zum Seminar.

1.4 Anmeldung

Wenn Sie am Seminar teilnehmen möchten, dann kommen Sie einfach zum Auftakt am 07. und 08. April dazu.

Sollte Ihnen eine Teilnahme an dieser Auftaktveranstaltung nicht möglich sein, senden Sie eine E-Mail an: volker.ulm@uni-bayreuth.de

1.5 Prüfungsleistung

Die Prüfungsform ist „Präsentation“ gemäß § 11 Abs. 10 der Prüfungsordnungen für MINT-Lehramt PLUS an der Universität Bayreuth.

1.6 Ansprechpartner

Generell Ansprechpartner für das Seminar ist:

Prof. Dr. Volker Ulm, volker.ulm@uni-bayreuth.de, <https://www.dmi.uni-bayreuth.de>

Speziell bei Fragen zu Python bzw. zu Informatik ist behilflich:

Andreas Walter, andreas.walter@uni-bayreuth.de, <https://did.inf.uni-bayreuth.de>

1.7 Fahrplan zum Seminar

Der folgende Arbeits- und Zeitplan weist einen Weg durch die Kapitel 2, 3 und 4 dieses Skripts. Er beschreibt, was die Seminarteilnehmer bis zum jeweiligen Treffen für den gemeinsamen Austausch gemacht haben sollten.

Dieser Plan kann sich noch im Verlauf des Seminars je nach den Wünschen der Teilnehmer entwickeln.

bis Treffen 2, 08. April 2021	Python installieren Installieren Sie eine Entwicklungsumgebung für Python (siehe Abschnitt 2.1.2), z. B. https://thonny.org
bis Treffen 3, 30. April 2021	Einstieg in Python Erarbeiten Sie sich einen Einstieg in Python mit einem Online-Text-Tutorial (Abschnitt 2.2.1), einem Online-Video-Tutorial (Abschnitt 2.2.2) oder einem Lehrbuch (Abschnitt 2.2.3). Orientieren Sie sich dabei an den Inhalten, die auf den ersten beiden Seiten von Abschnitt 2.2 genannt sind. Arbeiten Sie selbstorganisiert untereinander zusammen. Als Ansprechpartner stehen Ihnen die in Abschnitt 1.6 genannten Personen zur Verfügung. Mathematik und Programmieren Verbinden Sie den Einstieg in Python mit dem Erstellen von Programmen zu den mathematischen Inhalten aus den Abschnitten 3.1 bis 3.3.
bis Treffen 4, 21. Mai 2021	Einstieg in Python Vertiefen Sie Ihren Einstieg in Python, indem Sie mit einem Online-Text-Tutorial (Abschnitt 2.2.1), einem Online-Video-Tutorial (Abschnitt 2.2.2) oder einem Lehrbuch (Abschnitt 2.2.3) weiterarbeiten. Orientieren Sie sich dabei an den Inhalten, die auf der ersten Seite von Abschnitt 2.2 genannt sind. Arbeiten Sie selbstorganisiert untereinander zusammen. Als Ansprechpartner stehen Ihnen die in Abschnitt 1.6 genannten Personen zur Verfügung. Mathematik und Programmieren Erstellen Sie Programme zu den mathematischen Inhalten aus Abschnitt 3.4. Fachdidaktik Lesen Sie die beiden in Abschnitt 4.1 näher beschriebenen Texte „Was ist algorithmisches Denken“ (4 Seiten) und „Wenn ..., dann ... bis ..., Algorithmisches Denken (nicht nur) im Mathematikunterricht“ (8 Seiten) und reflektieren Sie sie.
bis Treffen 5, 11. Juni 2021	Mathematik und Programmieren Erstellen Sie Programme zu den mathematischen Inhalten aus den Abschnitten 3.5 und 3.6. Fachdidaktik Reflektieren Sie die von Ihnen bislang genutzten Lernangebote zum Einstieg in das Programmieren mit Python im Hinblick auf Ihr eigenes Lernen und auf möglichen Nutzen für Schüler. Was war gut? Was war schlecht? Warum? Lesen Sie gemäß Abschnitt 4.2 im Buch „Mathematische Algorithmen im Unterricht“ von Oldenburg das Kapitel 1 „Didaktische Ziele“ (10 Seiten) und setzen Sie sich mit den dargestellten Standpunkten auseinander.

bis Treffen 6,
25. Juni 2021

Mathematik und Programmieren

Erstellen Sie Programme zu den mathematischen Inhalten aus Abschnitt 3.7.

Fachdidaktik und Mathematik

Schmökern Sie gemäß Abschnitt 4.3 im Buch „Mathematische Algorithmen im Unterricht“ von Oldenburg.

Denken Sie über die Thematik nach, ob und wie Schüler im Schulalltag einen Einstieg in mathematisches Arbeiten mit einer Programmiersprache finden sollten bzw. können. Denken Sie dabei über organisatorische, methodische Fragen nach, aber auch über Ziele des Mathematikunterrichts und Aspekte der individuellen Förderung und der Differenzierung.

bis Treffen 7,
09. Juli 2021

Mathematik und Programmieren

Erstellen Sie Programme zu den mathematischen Inhalten aus den Abschnitten 3.8 bis 3.11.

Fachdidaktik und Mathematik

Schmökern Sie gemäß Abschnitt 4.4 im Buch „Algorithmen von Hammurapi bis Gödel“ von Ziegenbalg et al.

2 Einstieg in Python

Welche Programmiersprache Sie für das Programmieren nutzen, ist im Grunde egal. Für das Folgende wurde die Sprache Python gewählt, denn:

- Python gehört zu den verbreitetsten und gefragtesten professionellen Programmiersprachen weltweit (Stand 2021).
- Für didaktische Zwecke und einen Einstieg ins Programmieren ist diese Sprache sehr gut geeignet, da sie mit wenigen Schlüsselwörtern auskommt und der Code ohne viel „Ballast“ sehr prägnant ist.
- Entwicklungsumgebungen zum Programmieren sind kostenfrei verfügbar – sowohl für die professionelle Nutzung als auch für das Lernen von Schülerinnen und Schülern.
- Es gibt umfangreiche, kostenfreie Angebote an Tutorials und didaktischen Materialien – insbesondere auch für Schülerinnen und Schüler.
- Python kann mit beliebig großen ganzen Zahlen exakt rechnen – ohne Begrenzungen der Stellenzahl.

Dieses Kapitel zeigt Wege, um die Programmiersprache Python kennenzulernen und Fähigkeiten zum Programmieren zu entwickeln.

Verbinden Sie den Einstieg in Python mit der selbstständigen Entwicklung von Programmen zu Inhalten der Schulmathematik. Hierzu finden Sie in Kapitel 3 zahlreiche Anregungen.

2.1 Technische Vorbereitungen

2.1.1 Online Interpreter

Für ein erstes Kennenlernen kann man Python ganz ohne Installationen im Internet-Browser nutzen. Beispielsweise bietet die ETH Zürich folgende Seite, die insbesondere für Schüler entwickelt wurde:

<https://webtigerjython.ethz.ch/>

Eine Internet-Recherche zu „python online“ führt auf weitere ähnliche Angebot wie etwa:

<https://www.programiz.com/python-programming/online-compiler/>

<http://www.python-online.ch/pyonline/PyOnline.php>

2.1.2 Integrierte Entwicklungsumgebung

Für systematischeres Arbeiten mit Python ist die Installation einer sog. „Integrierten Entwicklungsumgebung“ (Integrated Development Environment IDE) sinnvoll. Sie bietet einen Editor mit einem integrierten Python-Interpreter. Eine solche Software ist „Thonny“; sie wurde an der Universität Tartu (Estland) für Unterrichtszwecke entwickelt und steht auf folgender Seite kostenlos zum Download zur Verfügung:

<https://thonny.org>

Für das Folgende wird empfohlen, Thonny oder eine andere Integrierte Entwicklungsumgebung zu installieren. Weitere Entwicklungsumgebungen (z. B. [Spyder](#), [PyCharm](#)) sind etwa auf folgender Seite zusammengestellt:

<https://wiki.python.org/moin/IntegratedDevelopmentEnvironments>

2.2 Programmieren lernen und Lehr-Lern-Medien didaktisch bewerten

Es gibt eine Vielfalt und Fülle an Medienangeboten, um Python zu lernen. Im Folgenden wird nur ein kleiner Ausschnitt dargestellt. Gegliedert ist dies in die Kategorien

- Online-Text-Tutorials (Abschnitt 2.2.1),
- Online-Video-Tutorials (Abschnitt 2.2.2),
- Bücher, inkl. e-books (Abschnitt 2.2.3),
- Online-Schulungen (Abschnitt 2.2.4).

Suchen Sie sich hier ein Angebot aus, das Ihnen passend erscheint. Wenn Sie damit einen Einblick in das Programmieren mit Python gewonnen haben, sehen Sie sich auch noch andere Angebote an.

Betrachten und bewerten Sie die Lernmedien zur Einführung in Python auch unter didaktischen Gesichtspunkten.

- Inwiefern sind die Angebote für Ihr eigenes Lernen günstig oder ungünstig gestaltet?
- Wie schätzen Sie die Angebote in Bezug auf das Lernen von Schülern ein?

Vorschlag: Elemente von Python

Angesichts der Fülle an Möglichkeiten, die Python bietet, können Sie sich inhaltlich beschränken. Um substanzielle Probleme zur Schulmathematik der Sekundarstufe numerisch zu bearbeiten, genügt es, sich mit folgenden Aspekten von Python vertraut zu machen:

Variablen und Zuweisungen (z. B. „a = 5“ oder „a = b + c“)

Datentypen

- ganze Zahlen: int
- Dezimalzahlen: float
- Texte: string
- Wahrheitswerte: bool
- evtl. Listen

Rechenoperation

- für int-Zahlen
 - +, -, *
 - // (Ganzzahldivision, z. B. 17 // 5 = 3)
 - % (Rest bei Ganzzahldivision, z. B. 17 // 5 = 2)
 - ** (Potenzieren)
- für float-Zahlen
 - +, -, *, /
 - ** (Potenzieren)

Eingabe/Ausgabe

- input(): Ergebnis der Eingabe ist ein string, Umwandlung in Zahlen mit int() und float()
- print(): Ausgabe

Vergleiche

- ==, != (gleich, ungleich)
- <, <=, >, >=

Bedingungen

- if, elif, else

Logische Operatoren

- and, or, not

Schleifen

- while
- for (hier nützlich: range(n) liefert Liste der Zahlen 0, 1, ..., n-1,
allgemeiner: range(a, b, t) liefert Liste der Zahlen von a bis b-1 in der Schrittweite t)

Funktionen

- ohne/mit Übergabe von Parametern
- ohne/mit Rückgabewert

Mathematische Funktionen aus der Standardbibliothek

- import math

Verbinden Sie den Einstieg in Python mit dem Erstellen eigener Programme. Kapitel 3 gibt hierzu inhaltliche Anregungen für Programme zu Inhalten der Schulmathematik.

2.2.1 Online-Text-Tutorials

Für den Einstieg ins Programmieren und in Python

<https://pythonbuch.com>

Die Autoren dieser Website schreiben zu Beginn:

„Dieses Skript wurde hauptsächlich für eine Einleitung ins Programmieren mit Python auf Gymnasiumsstufe geschrieben. Es richtet sich an Schülerinnen und Schüler eines Grundkurses, welche noch keine Programmiererfahrung haben. Es wurde darauf geachtet, dass auch nicht Computer-affine Personen dem Inhalt folgen können und davon profitieren. In dem Sinne ist es besser für einen Grundkurs als einen Leistungskurs geeignet.

Es ist so verfasst, dass man dem Skript schrittweise folgen und ohne vorherige Einführung durch die Lehrperson erste Aufgaben lösen kann. Es wurde für selbständiges Arbeiten der Schülerinnen und Schüler geschrieben. Es ist daher auch für Personen geeignet, welche sich selbständig erste Kenntnisse im Programmieren aneignen möchten.“

<https://www.snakify.org/de/>

Diese Website bietet einen klaren Einstieg in Python (in Englisch, Deutsch und anderen Sprachen). Dies ist in „Lessons“ gegliedert, die Lerninhalte finden sich auf den sog. „Theory“-Seiten. Dabei wird viel mit Beispielen gearbeitet; die zugehörigen Python-Programme sind auf der jeweiligen Seite direkt im Browser ausführbar. Man kann mit dem Angebot intensiv arbeiten, auch ohne sich als Benutzer anzumelden (ohne „Sign up“). Nur falls man Lösungen zu den Übungsaufgaben haben möchte, wird eine Anmeldung als Benutzer gefordert.

Wenn man schon etwas Erfahrung mit Programmieren hat

Auch die folgenden Seiten sind Einführungen in Python. Im Gegensatz zu den im vorherigen Abschnitt genannten Seiten sind sie etwas ausführlicher und gehen etwas mehr in die Tiefe:

<https://www.python-lernen.de>

https://www.python-kurs.eu/python3_kurs.php

<https://www.learnpython.org>

<https://py-tutorial-de.readthedocs.io>

<https://www.w3schools.com/python/>

2.2.2 Online-Video-Tutorials

Eine Internetrecherche führt zu zahlreichen Videos, in denen eine Einführung in Python gegeben wird.

Das folgende Youtube-Video wurde über 23 Millionen Mal geklickt (Stand 01.04.2021), es ist eine insgesamt vierstündige Einführung in Python:

<https://www.youtube.com/watch?v=rfscVS0vtbw>

Eine einstündige Einführung wird hier angeboten:

<https://www.youtube.com/watch?v=kqtD5dpm9C8>

Die gleiche Person bietet auch eine sechsstündige Einführung, wobei die erste Hälfte für einen Einstieg in Python ausreichend ist:

https://www.youtube.com/watch?v=_uQrJ0TkZlc

2.2.3 Bücher

Wer gern ein gedrucktes Buch liest, findet eine umfangreiche Auswahl an Literatur zur Einführung in Python. Für einen Einstieg in Python genügt es in der Regel, nur einige Kapitel dieser Bücher zu lesen. Beispiele sind:

Bonacina, Michael (2018): Python 3, Programmieren für Einsteiger, BMU Media, Landshut
(UBT-Signatur: 80 ST 250 P 99 B 697)

Theis, Thomas (2020): Einstieg in Python, Rheinwerk Verlag, Bonn
(UBT-Signatur: 80 ST 250 P 99 T 377 (6))

Downey, Allan (2014): Programmieren lernen mit Python, O'Reilly, Köln
(UBT-Signatur: 80 ST 250 P 99 D 748)

Die folgenden Bücher können über die Universitätsbibliothek als PDF heruntergeladen werden (z. B. über <https://link.springer.com/athens-shibboleth-login>):

Dörn, Sebastian (2019): Python lernen in abgeschlossenen Lerneinheiten, Programmieren für Einsteiger mit vielen Beispielen, Springer Vieweg, Wiesbaden
<https://link.springer.com/book/10.1007/978-3-658-26496-3>

Steyer, Ralph (2018): Programmierung in Python, Ein kompakter Einstieg für die Praxis, Springer Vieweg, Wiesbaden
<https://link.springer.com/book/10.1007/978-3-658-20705-2>

Hunt, John (2020): A Beginners Guide to Python 3 Programming, Springer, Cham
<https://link.springer.com/book/10.1007/978-3-030-20290-3>

Speziell für Physiker ist folgendes Buch geschrieben, wobei das Kapitel „2 Einführung in Python“ (60 Seiten) nicht für Physik spezifisch ist:

Natt, Oliver (2020): Physik mit Python, Simulationen, Visualisierungen und Animationen von Anfang an, Springer Spektrum, Berlin
<https://link.springer.com/book/10.1007/978-3-662-61274-3>
<https://pyph.de>

2.2.4 Online-Schulungen

Es gibt einen umfangreichen Markt für Software-Schulungen (mit Vorlesungen, Tutorials, Übungsaufgaben, ...). Da man sich für die Kurse in der Regel einschreiben muss und sie oft kostenpflichtig sind, werden sie im Rahmen dieses Seminars nicht weiter betrachtet. Anbieter von Python-Kursen im Internet sind beispielsweise:

www.udemy.com, www.datacamp.com, www.coursera.org, www.edx.org, www.codecademy.com

Ein vierwöchiger, deutschsprachiger Kurs speziell für Schülerinnen und Schüler ist:

<https://open.hpi.de/courses/pythonjunior2020>

3 Mathematische Verfahren erschließen, Algorithmen und Programme entwickeln und nutzen

Programmieren lernt man durch Programmieren. Erstellen Sie begleitend zu Ihrem Einstieg in Python selbst Programme, die Berechnungen zu Inhalten der Schulmathematik durchführen. Dies umfasst vielfältige Anforderungen und fördert entsprechende Kompetenzen:

- mathematische Inhalte erschließen,
- mathematische Verfahren entwickeln,
- Algorithmen zu den Verfahren entwickeln,
- Programme zu den Algorithmen verfassen,
- Programme testen und bewerten.

In diesem Kapitel finden Sie hierfür mögliche Inhalte der Schulmathematik.

Die Themen sind in zwei Anforderungsstufen differenziert:

- Blau eingerahmte Themen sind mit eher kurzen, einfachen Programmen bearbeitbar. Sie dienen insbesondere zum Einstieg in das Programmieren und das Kennenlernen einer Programmiersprache.
- Orange eingerahmte Themen erfordern etwas komplexere Programme, sind aber auch im Rahmen dessen, was man als Programmierneinsteiger machen kann.

Stellen Sie sich auch selbst weitere Programmieraufgaben zu selbst gewählten Inhalten der (Schul-) Mathematik.

Diskutieren Sie dabei auftretende Schwierigkeiten, gewonnenen Ergebnisse, ... in der Gruppe aller Teilnehmer.

(Zu den obigen Begriffen:

- Ein „Algorithmus“ ist eine endliche Folge von eindeutig bestimmten Elementaranweisungen, die den Lösungsweg eines Problems exakt und vollständig beschreiben.
- Ein „Programm“ ist eine Formulierung eines Algorithmus mit einer Programmiersprache.)

3.1 Große natürliche Zahlen

Zweierpotenzen

Einer alten Legende nach wollte ein König den Erfinder des Schachspiels für seine Erfindung belohnen. Der Erfinder bat den König: Legt auf das erste Feld des Schachbretts ein Reiskorn, auf das zweite Feld zwei Reiskörner, auf das dritte Feld vier Körner, auf das vierte Feld acht Körner und so weiter. Auf jedem Feld sollen doppelt so viele Reiskörner liegen wie auf dem vorhergehenden. Gebt mir dann bitte den Reis auf allen 64 Feldern des Schachbretts. Der König war erzürnt, weil er den Wunsch für zu bescheiden hielt.

Erstellen Sie ein Programm, das berechnet, wie viele Reiskörner auf den einzelnen Feldern liegen müssten und viele Reiskörner insgesamt auf dem Schachbrett wären.

Vergleichen Sie das Ergebnis mit der Jahresproduktion an Reis auf der Erde.

3.2 Kombinatorik

Permutationen (ohne Wiederholungen, mit Berücksichtigung der Reihenfolge)

Wie viele Möglichkeiten gibt es, $n = 10$ Musikstücke nacheinander anzuhören?

Erstellen Sie zur Berechnung ein Programm, bei dem man den Wert von n eingeben kann.

Variationen (ohne Wiederholungen, mit Berücksichtigung der Reihenfolge)

Es stehen $n = 100$ Musikstücke zur Auswahl. Wie viele Möglichkeiten gibt es, hiervon $k = 10$ Musikstücke nacheinander anzuhören?

Erstellen Sie zur Berechnung ein Programm, bei dem man die Werte von n und k eingeben kann.

Kombinationen (ohne Wiederholungen, ohne Berücksichtigung der Reihenfolge)

Es stehen $n = 100$ Musikstücke zur Auswahl. Wie viele Möglichkeiten gibt es, hiervon $k = 10$ Musikstücke zu kaufen? (Auf die Reihenfolge der 10 Musikstücke kommt es beim Kauf nicht an.)

Erstellen Sie zur Berechnung ein Programm, bei dem man die Werte von n und k eingeben kann.

3.3 Teiler und Vielfache

Größter gemeinsamer Teiler

Wie lautet der größte gemeinsame Teiler von 36043 und 77777?

Recherchieren Sie zum Euklidischen Algorithmus für den ggT.

Erstellen Sie ein Programm, das für zwei einzugebende Zahlen den ggT bestimmt.

Kleinstes gemeinsames Vielfaches

Wie lautet das kleinste gemeinsame Vielfache von 36043 und 77777?

Begründen Sie anhand der Primfaktorzerlegungen zweier natürlicher Zahlen a und b :

$$\text{ggT}(a, b) \cdot \text{kgV}(a, b) = a \cdot b$$

Erstellen Sie ein Programm, das für zwei einzugebende Zahlen das kgV bestimmt.

Häufigkeiten teilerfremder Zahlen

Im kartesischen Koordinatensystem betrachten wir für $n \in \mathbb{N}$ die Menge aller Punkte (x, y) mit natürlichen Zahlen $1 \leq x, y \leq n$ als Koordinaten (also ein quadratisches Punktemuster).

Wie groß ist der Anteil der Punkte (x, y) , bei denen $\text{ggT}(x, y) = 1$ ist?

Wie verhält sich dieser Anteil für zunehmende n ?

Erstellen Sie ein Programm zur Erkundung dieser Thematik.

(Hinweis: Für $n \rightarrow \infty$ konvergiert der Anteil gegen $\frac{6}{\pi^2}$ (vgl. Engel 1991, S. 31).

Euler'sche φ -Funktion

Die Euler'sche φ -Funktion gibt für $n \in \mathbb{N}$ die Anzahl der natürlichen Zahlen an, die kleiner oder gleich n sind und mit n den größten gemeinsamen Teiler 1 besitzen.

$$\varphi(n) = |\{i \in \mathbb{N} | 1 \leq i \leq n \wedge \text{ggT}(i, n) = 1\}|$$

Erstellen Sie ein Programm zur Berechnung dieser Funktion.

(Hinweis: Die φ -Funktion ist beispielsweise von Bedeutung bei Verschlüsselungen mit dem RSA-Verfahren bzw. generell in der Algebra und Zahlentheorie. $\varphi(n)$ ist die Anzahl der invertierbaren Elemente des Rings $\mathbb{Z}/(n)$.)

3.4 Primzahlen

Primzahltest

Ist die Zahl 23456789 eine Primzahl?

Erstellen Sie ein Programm, das für eine einzugebende Zahl feststellt, ob sie prim ist.

Primzahlliste

Welche Primzahlen gibt es bis zu 10 000?

Erstellen Sie ein Programm, das alle Primzahlen bis zu einer einzugebenden Zahl ausgibt.

Primfaktorzerlegung

Wie lautet die Primfaktorzerlegung von 123456789?

Erstellen Sie ein Programm, das die Primfaktorzerlegung einer einzugebenden Zahl ausgibt.

Goldbach'sche Vermutung

Die Goldbach'sche Vermutung wurde vom Mathematiker Christian Goldbach (1690 – 1764) im Jahr 1742 in einem Brief an Leonhard Euler (1707 – 1783) formuliert. Sie ist bis heute weder bewiesen noch widerlegt und gehört zu den bekanntesten ungelösten Problemen der Mathematik:

Jede gerade Zahl, die größer als 2 ist, ist Summe zweier Primzahlen.

Erstellen Sie ein Programm, das bei Eingabe einer geraden Zahl Darstellungen als Summe zweier Primzahlen bestimmt.

Primzahlzwillinge

Ein Paar $(n; n + 2)$ natürlicher Zahlen heißt *Primzahlzwillig*, wenn n und $n + 2$ Primzahlen sind.

Erstellen Sie ein Programm, um Primzahlzwillinge zu bestimmen.

Primzahldrillinge

Analog zum vorhergehenden könnte man einen Primzahldrilling definieren als Tripel $(n; n + 2; n + 4)$, bei dem die drei Zahlen Primzahlen sind.

Begründen Sie, dass es bei dieser Definition nur einen einzigen Primzahldrilling gäbe.

Deshalb definiert man einen *Primzahldrilling* als Tripel $(n; n + 2; n + 6)$ oder $(n; n + 4; n + 6)$, bei dem die drei Zahlen Primzahlen sind.

Erstellen Sie ein Programm, um Primzahldrillinge zu bestimmen.

3.5 Nullstellen von Funktionen

Intervallhalbierungsverfahren

Welche Nullstelle hat die Funktion $f(x) = \cos(x) - x$?

Recherchieren Sie zum Intervallhalbierungsverfahren für die Bestimmung von Nullstellen von Funktionen.

Erstellen Sie ein Programm, mit dem Sie mit diesem Verfahren Nullstellen berechnen können.

Sekantenverfahren

Für welche Zahl ist $x^x = 2$?

Recherchieren Sie zum Sekantenverfahren für die Bestimmung von Nullstellen von Funktionen.

Erstellen Sie ein Programm, mit dem Sie mit diesem Verfahren Nullstellen berechnen können.

Tangentenverfahren

Welche Lösung hat die Gleichung $x^5 - x - 1 = 0$?

Recherchieren Sie zum Tangentenverfahren (= Newtonverfahren) für die Bestimmung von Nullstellen von Funktionen.

Erstellen Sie ein Programm, mit dem Sie mit diesem Verfahren Nullstellen berechnen können.

3.6 Wurzeln

Heron-Verfahren

Wie lassen sich Quadratwurzeln durch iteriertes Ausführen von Grundrechenarten berechnen?

Recherchieren Sie zum Heron-Verfahren.

Interpretieren Sie das Verfahren geometrisch (Rechtecke nähern sich einem Quadrat bei gleichbleibendem Flächeninhalt an) und mit Analysis (Nullstellen mit dem Newton-Verfahren).

Erstellen Sie ein Programm, das mit dem Heron-Verfahren die Quadratwurzel eingegebener Zahlen berechnet.

Verallgemeinerung des Heron-Verfahren

Wie lassen sich n -te Wurzeln durch iteriertes Ausführen von Grundrechenarten berechnen?

Verallgemeinern Sie das Heron-Verfahren auf die Berechnung von $\sqrt[n]{a}$, indem Sie das Newton-Verfahren auf die Funktion $f(x) = x^n - a$ anwenden.

Erstellen Sie ein Programm, das $\sqrt[n]{a}$ für eingegebene Zahlen n und a berechnet.

3.7 Kreiszahl π

3.7.1 Mit Geometrie π berechnen

Verfahren von Archimedes

Recherchieren Sie zum Verfahren von Archimedes zur Berechnung von π .

Zeigen Sie: Betrachtet man in einen Kreis mit Radius 1 einbeschriebene n -Ecke, so gilt für die Seitenlängen s_n die Rekursionsformel:

$$s_{2n} = \sqrt{2 - \sqrt{4 - s_n^2}}$$

Zeigen Sie, dass auch gilt:

$$s_{2n} = \frac{s_n}{\sqrt{2 + \sqrt{4 - s_n^2}}}$$

Erstellen Sie ein Programm, das mit dem Verfahren von Archimedes Näherungen für π berechnet.

Vergleichen Sie die Konvergenz dieser Näherungen bei Nutzung der obigen beiden Formeln.

Verfahren von Cusanus

Recherchieren Sie zum Verfahren von Cusanus zur Berechnung von π .

Erstellen Sie ein Programm, das mit dem Verfahren von Cusanus Näherungswerte von π berechnet.

3.7.2 Mit Analysis π berechnen

Die Analysis führt zu sehr vielfältigen Darstellungen von π , beispielsweise als

- unendliche Summe:

$$\frac{\pi^2}{6} = 1 + \frac{1}{4} + \frac{1}{9} + \frac{1}{16} + \frac{1}{25} + \dots$$

Leonhard Euler, 1739

$$\frac{1}{\pi} = \frac{2\sqrt{2}}{9801} \sum_{n=0}^{\infty} \frac{(4n)!}{(n!)^4} \cdot \frac{26390n+1103}{396^{4n}}$$

Srinivasa Ramanujan, 1910

- unendliches Produkt:

$$\frac{\pi}{2} = \frac{2 \cdot 2}{1 \cdot 3} \cdot \frac{4 \cdot 4}{3 \cdot 5} \cdot \frac{6 \cdot 6}{5 \cdot 7} \cdot \frac{8 \cdot 8}{7 \cdot 9} \cdot \dots$$

John Wallis, 1655

$$\frac{2}{\pi} = \sqrt{\frac{1}{2}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}} \cdot \sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2} + \frac{1}{2}\sqrt{\frac{1}{2}}}} \cdot \dots$$

Francois Viète (Vieta), 1579

- Integral:

$$\pi = \int_{-\infty}^{\infty} \frac{1}{x^2+1} dx$$

$$\pi = \int_{-1}^1 \frac{1}{\sqrt{1-x^2}} dx$$

Im Folgenden wird ausgehend von der Potenzreihenentwicklung des Arcustangens ein leistungsfähiges Verfahren zur Berechnung vieler Stellen von π entwickelt.

Die Arcustangens-Reihe (Teil 1)

Auf James Gregory (1638 – 1675) geht folgende Reihe zurück:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \dots$$

Begründen Sie dies mit der Potenzreihenentwicklung des Arcustangens:

$$\arctan x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1} \quad \text{für } |x| \leq 1$$

Vergewissern Sie sich, dass die obige Reihe ungünstig zur Berechnung von π ist.

Die Arcustangens-Reihe (Teil 2)

Eine schnelle Konvergenz der Potenzreihenentwicklung des Arcustangens

$\arctan x = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{2n+1}$ erhält man, wenn $|x|$ deutlich kleiner als 1 ist. (Warum?)

Damit führt der folgende Zusammenhang zu einem leistungsstarken Verfahren zur Berechnung von π :

$$\arctan(1) = 4 \cdot \arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right)$$

(Dies kann man beispielsweise anhand der Additionstheoreme aus der Trigonometrie beweisen oder indem man in der Gleichung $(5+i)^4 \cdot (239-i) = 114244 \cdot (1+i)$ auf beiden Seiten die Argumente der komplexen Zahlen betrachtet.)

Es ist also:

$$\frac{\pi}{4} = 4 \cdot \arctan\left(\frac{1}{5}\right) - \arctan\left(\frac{1}{239}\right)$$

Für die Werte $x = \frac{1}{5}$ und $x = \frac{1}{239}$ konvergiert die Potenzreihenentwicklung des Arcustangens recht schnell.

Erstellen Sie ein Programm, das damit π berechnet.

Viele Stellen von π

Mit Python kann die Zahl π auch auf viele Stellen genau berechnet werden (z. B. auf 1000 Stellen).

Mit Float-Zahlen wird in Python mit einer Genauigkeit von etwa 16 Stellen gerechnet. Bei Int-Zahlen gibt es eine solche Stellenbeschränkung allerdings nicht.

Wenn man π auf s Stellen genau berechnen will, kann man etwa $\pi \cdot 10^s$ als Int-Zahl berechnen:

$$\pi \cdot 10^s = 16 \cdot 10^s \cdot \arctan\left(\frac{1}{5}\right) - 4 \cdot 10^s \cdot \arctan\left(\frac{1}{239}\right)$$

Hierbei kann man bei den Potenzreihenentwicklungen ausschließlich mit Int-Zahlen rechnen:

$$10^s \cdot \arctan\left(\frac{1}{n}\right) = \frac{10^s}{n \cdot 1} - \frac{10^s}{n^3 \cdot 3} + \frac{10^s}{n^5 \cdot 5} - \frac{10^s}{n^7 \cdot 7} + \dots$$

Berechnen Sie damit die Kreiszahl π auf 1000 Stellen genau.

3.7.3 Mit Stochastik π berechnen

Monte-Carlo-Verfahren

Recherchieren Sie zum Monte-Carlo-Verfahren zur Berechnung von π .

Erstellen Sie ein Programm, das mit dem Monte-Carlo-Verfahren Näherungen für π berechnet.

Beurteilen Sie die Konvergenz dieses Verfahrens im Vergleich zu anderen Verfahren.

Hinweis: Im Python-Modul „random“ liefert die Funktion „random()“ Zufallszahlen aus $[0; 1[$.

```
import random
x = random.random()
```

3.8 Eulersche Zahl e

Reihendarstellung von e

Für die Eulersche Zahl gibt es die Reihendarstellung:

$$e = \sum_{k=0}^{\infty} \frac{1}{k!}$$

Berechnen Sie die Summe der ersten n Summanden und beobachten Sie, wie diese Summe mit zunehmendem n konvergiert.

Viele Stellen von e

Mit Python kann die Zahl e auch auf viele Stellen genau berechnet werden (z. B. auf 1000 Stellen).

Mit Float-Zahlen wird in Python mit einer Genauigkeit von etwa 16 Stellen gerechnet. Bei Int-Zahlen gibt es eine solche Stellenbeschränkung allerdings nicht.

Wenn man e auf s Stellen genau berechnen will, kann man etwa $e \cdot 10^s$ als Int-Zahl berechnen. Hierzu lässt sich die Reihendarstellung nutzen, indem man auf der rechten Seite ausschließlich mit Int-Zahlen rechnet:

$$e \cdot 10^s = \sum_{k=0}^{\infty} \frac{10^s}{k!}$$

Berechnen Sie damit die Eulersche Zahl e auf 1000 Stellen genau.

3.9 Numerische Integration

Verfahren zur numerischen Integration

Es ist durchaus normal, dass man zu einer etwas komplizierteren Funktion keine Stammfunktion explizit angeben kann, um damit bestimmte Integrale zu berechnen.

$$\int_0^1 \sqrt{x^3 + 1} dx, \quad \int_0^\pi \sqrt{\sin(x)} dx, \quad \int_0^1 \cos(2^x) dx, \quad \int_0^{2\pi} 2^{\sin(x)} dx, \quad \int_0^1 x^x dx, \quad \int_0^1 \sqrt{x} dx$$

Allerdings kann man numerische Verfahren nutzen, um Näherungswerte für Integrale zu berechnen.

Recherchieren Sie nach numerischen Verfahren zum Berechnen bestimmter Integrale („numerische Integration“). Beispiele sind:

- Rechteckregel,
- Sehnen-Trapez-Regel,
- Tangenten-Trapez-Regel,
- Simpson-Regel,
- Monte-Carlo-Integration.

Setzen Sie solche Verfahren mit einem Programm um.

Bestimmen Sie damit Näherungen für bestimmte Integrale.

Vergleichen Sie verschiedene Verfahren zur numerischen Integration bzgl. Vor- und Nachteilen.

3.10 Simulation von Zufallsexperimenten

Abstand zweier Punkte in einem Quadrat

In einem Quadrat mit Seitenlänge 1 werden wiederholt zwei Punkte zufällig gewählt. Wie weit sind sie im Schnitt voneinander entfernt?

Erstellen Sie zur Simulation ein Programm.

Bei Interesse können Sie die Häufigkeitsverteilung auch noch differenzierter untersuchen.

(Hinweis: Der Erwartungswert beträgt $\frac{2+\sqrt{2}+5 \cdot \ln(1+\sqrt{2})}{15}$ (vgl. Engel 1991, S. 93).

Abstand zweier Punkte in einem Kreis

In einem Kreis mit Seitenlänge 1 werden wiederholt zwei Punkte zufällig gewählt. Wie weit sind sie im Schnitt voneinander entfernt?

Erstellen Sie zur Simulation ein Programm.

Bei Interesse können Sie die Häufigkeitsverteilung auch noch differenzierter untersuchen.

(Hinweis: Der Erwartungswert beträgt $\frac{128}{45\pi}$ (vgl. Engel 1991, S. 94).

Abstand zweier Punkte auf einem Kreis

Auf einem Kreis mit Seitenlänge 1 werden wiederholt zwei Punkte zufällig gewählt. Wie weit sind sie im Schnitt voneinander entfernt?

Erstellen Sie zur Simulation ein Programm.

Suchen Sie einen Weg zur Berechnung des Erwartungswerts mit Mitteln der Schulmathematik.

Tipp:

<https://www.spiegel.de/karriere/ein-kreis-zwei-punkte-raetsel-der-woche-a-daae881f-621f-44ca-8e5e-0000df47e481>

(Der Erwartungswert beträgt $\frac{4}{\pi}$).

3.11 Weitere mathematische Inhalte für Algorithmen

Suchen Sie selbst mathematische Inhalte, die sich mit Algorithmen bearbeiten lassen und setzen Sie dies an einem Computer um. Beispielsweise sind Stichworte für mögliche Inhalte:

- Darstellung von Zahlen in verschiedenen Zahlssystemen und Umwandlung von Zahldarstellungen
- Fibonacci-Zahlen
- Armstrong-Zahlen
- Sieb des Eratosthenes
- Pythagoreische Zahltripel
- Pascal'sches Dreieck
- Lösen von linearen Gleichungssystemen, Gauß-Algorithmus
- Rechnen mit Matrizen
- Simulation von Zufallsexperimenten, Verteilungen, statistische Kennzahlen
- Kryptographie, Verschlüsselungsverfahren, RSA-Algorithmus
- Mandelbrot-Menge, Julia-Mengen, Feigenbaum-Diagramm
- Interpolation
- Populationsentwicklungen (z. B. Räuber-Beute-Modell, logistische Gleichung)
- Sitzverteilung nach Wahlen (z. B. Verfahren von d'Hondt, Hare-Niemeyer)
- ...

4 Fachdidaktische Impulse

Die im folgenden genannten Lehrbücher im Springer Verlag können über die Universitätsbibliothek als PDF heruntergeladen werden (z. B über <https://link.springer.com/athens-shibboleth-login>). Weitere Materialien erhalten die Seminarteilnehmer per E-Mail.

4.1 Algorithmisch denken

Im Material zum Seminar finden Sie das Dokument „Was ist algorithmisches Denken.pdf“ (4 Seiten). Lesen Sie dieses Dokument, reflektieren Sie es und stellen Sie Bezüge zum Mathematikunterricht her.

Die Zeitschrift „mathematik lehren“ hat primär Mathematiklehrkräfte als Zielgruppe. Das Heft Nr. 188 vom Februar 2015 besitzt den Titel „Algorithmen“. (UBT-Signatur: 84 SA 6315.1 - 2014/15)

Lesen Sie – zumindest – folgenden Basisartikel (8 Seiten) in diesem Heft und reflektieren Sie ihn. Er findet sich als PDF im Material zum Seminar.

Kortenkamp, U., Lambert, A. (2015): Wenn ..., dann ... bis ..., Algorithmisches Denken (nicht nur) im Mathematikunterricht, mathematik lehren, Heft 188, S. 2-9

4.2 Programmieren und Mathematikunterricht

Lesen Sie in folgendem Buch das Kapitel 1 „Didaktische Ziele“ (10 Seiten). Setzen Sie sich mit den in diesem Kapitel dargestellten Standpunkten auseinander und beziehen Sie selbst Stellung dazu.

Oldenburg, Reinhard (2012): Mathematische Algorithmen im Unterricht, Mathematik aktiv erleben durch Programmieren, Vieweg + Teubner, Wiesbaden
<https://link.springer.com/book/10.1007/978-3-8348-8336-0>

4.3 Mathematische Algorithmen im Unterricht

Das unter 4.2 genannte Buch bereits bietet eine sehr reichhaltige Darstellung von mathematischen Algorithmen, mit denen interessierte Schüler arbeiten könnten. Dies ist mit einer Einführung in Python verbunden. Schmökern Sie in dem Buch.

Oldenburg, Reinhard (2012): Mathematische Algorithmen im Unterricht, Mathematik aktiv erleben durch Programmieren, Vieweg + Teubner, Wiesbaden
<https://link.springer.com/book/10.1007/978-3-8348-8336-0>

4.4 Vertiefung zu Algorithmen

Das folgende Buch verbindet Mathematik und Informatik. Die Autoren schreiben im Vorwort: „Dieses Buch hat das Ziel, in die Grundlagen und die Geistesgeschichte des Algorithmierens einzuführen.“ Es ist insbesondere auch für Lehramtsstudierende und im Hinblick auf das Lehren und Lernen in der Schule geschrieben. Es eignet sich zur Vertiefung der Inhalte des Seminars und zum Schmökern.

Ziegenbalg, J., Ziegenbalg, O., Ziegenbalg, B. (2016): Algorithmen von Hammurapi bis Gödel, Springer Spektrum, Wiesbaden
<https://link.springer.com/book/10.1007/978-3-658-12363-5>

4.5 Mathematik mit Python

Das folgende Buch gibt in 69 Kapiteln eine Einführung in ein breites Spektrum an Themen der Schul- und Hochschulmathematik (z. B. ganze Zahlen, Vektoren, Grenzwerte, Integrale, Potenzreihen, Differentialgleichungen, Zufallsvariablen u.v.m.). Dies ist gleichzeitig im einer Einführung in Python verwoben; zu den Inhalten werden entsprechende Programmcodes besprochen.

Weitz, Edmund (2021): Konkrete Mathematik (nicht nur) für Informatiker, Mit vielen Grafiken und Algorithmen in Python, Springer Spektrum, Berlin
<https://www.springer.com/de/book/9783662626184>

4.6 Ältere fachdidaktische Literatur

Die folgenden beiden „älteren“ Bücher sind reichhaltige Fundgruben für mathematische Ideen zu algorithmischem Arbeiten im Mathematikunterricht (inkl. der Umsetzung in Programmcode). Sie sind z. B. in der Universitätsbibliothek Würzburg verfügbar.

Engel, Arthur (1991): Mathematisches Experimentieren mit dem PC, Ernst Klett Schulbuchverlag, Stuttgart

Engel, Arthur (1977): Elementarmathematik vom algorithmischen Standpunkt, Klett Studienbücher, Stuttgart