Ein paar Hilfen zum Überprüfen und Experimentieren

Der Fokus der Tools ist auf dem was ich unterrichte und forsche, also Algebra, lineare Algebra, geometrische Dinge, ein wenig Analysis etc.

0.0.1 Wann macht das Rechnen per Computer Sinn?

Rechnen mit dem Computer macht Sinn:

- zum Überprüfen von Einzelschritten
- zum Produzieren von Evidenz für Probleme
- zum Produzieren von Beispielen
- zum Malen von Mengen (oder Graphen)
- zum Rechnen von aufwendigen Dingen (große Matrizen, Serien von Matrizen,...)

Wann macht es keinen Sinn... to be continued

0.0.2 Übersicht

Lineare Algebra wird gerne entweder auf Abbildungen über $\mathbb{R}$, oder $\mathbb{C}$ Vektorräumen oder über endlichen Körpern, z.B. $\mathbb{F}_3, \mathbb{F}_5$ betrieben. Zum Überprüfen von Ergebnissen gibt es mehrere sehr leicht zugängliche Softwaretools.

- Für Matrizen über $\mathbb{R}$ und $\mathbb{C}$ empfehlen sich entweder **Wolfram|Alpha** - mit Online Version, kostenlos - oder **MATLAB** (auf den Unirechnern installiert); **Wolfram|Alpha** ist am einfachsten zu bedienen, kommt aber auch am schnellsten an seine Grenzen.
- Für Matrizen über endlichen Körpern empfiehlt sich **Singular** (aus Kaiserslautern). Alternativ gehen auch **SageMath** oder **Macaulay2** (beide mit Webinterface).

Insbesondere hiflt dies dabei schnell zu überprüfen ob Kerne, Ränge, Eigenvektoren und Eigenwerte richtig berechnet wurden.

Für Analysis bieten sich diese Tools ebenfalls an. Zum visualisieren von Dingen (in 2-3 Dimensionen) bietet sich inbesondere das Webinterface von **Desmos** an.

Hier ist eine kurze Beispielliste für einige Übungsaufgaben.

0.1 Desmos (nur Online)

- Wenn du mehrere Graphen zeichnen möchtest die maximal zweidimensional sind
- Hat auch mehrere Parameter Slider, du kannst also teilweise (mit ein wenig Vorstellungskraft) auch dreidimensionale Dinge zeichnen (oder vier,...)

Webinterface: www.desmos.com.

Hier ist ein einfaches Beispiel zu **Desmos**. Du hast eine Menge $M_b \subset \mathbb{R}^2$, beschrieben durch eine Gleichung, für jedes $b \in \mathbb{Z}$:

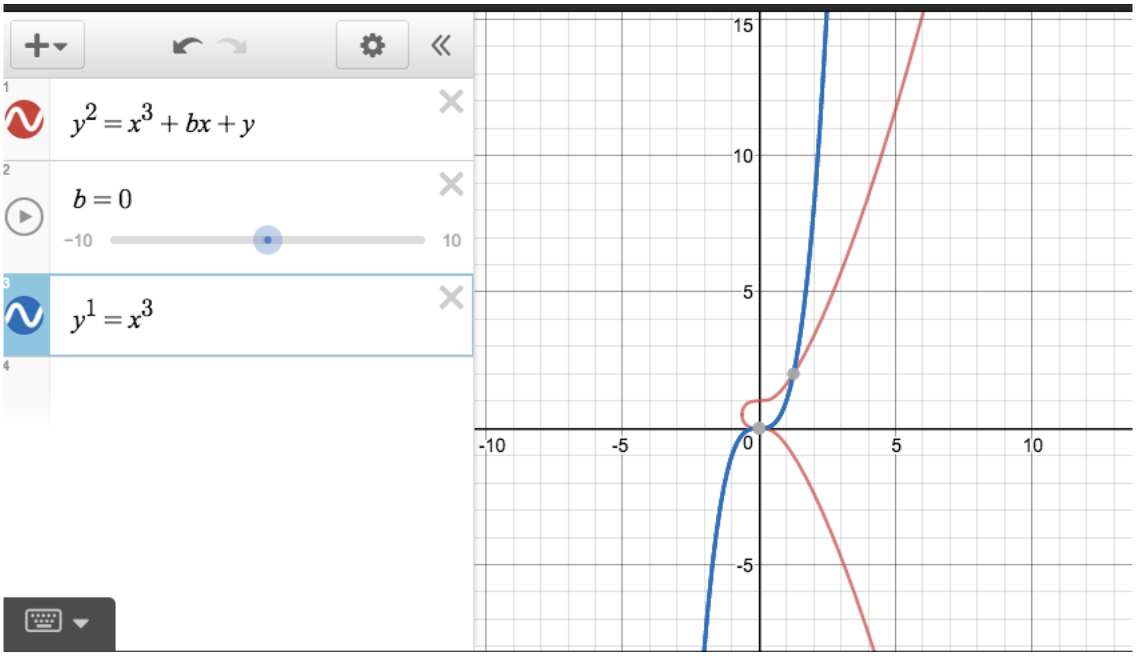
$$y^2 = x^3 + bx + y.$$

Dein Problem könnte sein, dass du die Nullstellen der Familie von Funktionen $f_b(x, y) = y - y^2 + x^3 + bx + y$, suchst. In diesem Fall könnte es dich interessieren, ob die Menge M_b eine oder zwei Komponenten hat, abhängig von b.

Oder es könnte sein, dass f_b die Lösung einer Differentialgleichung ist und du gerne wüsstest, ob f_b für alle b stetig ist, oder ob es "singulär" ist für einige b.

Du könntest dich auch dafür interessieren, wie der Schnitt von f_b mit einer anderen Familie aussieht, weil du ein System von Gleichungen lösen möchtest.

Eine Idee der Antwort kriegst du, annäherungsweise durch **Desmos**. Dazu ist ein Tool in diesem Fall auch da, um dir Evidenz/Beispiele/Intuition für die Lösung des Problems zu vermitteln. Der nächste Schritt wäre dann die Formulierung und der Beweis von z.B. "M_b ist zusammenhängend für $b > 0$".



0.2 Wolfram|Alpha (Online)

- Sehr einfache Bedienung
- Surface Plots (dreidimensional, eine mehr als **Desmos**)
- Die kostenfreie Variante ist extrem limitiert durch die Inputgröße; Eine 11×11 Matrix wird schwierig

Webinterface: www.wolframalpha.com

Berechnet euch schnell charakteristische Polynome, Jordan Normalformen, Eigenvektoren etc. für $\mathbb{R}$ und $\mathbb{C}$ als Basiskörper. Einige Beispielbefehle:

```
{{2,-1,1},{0,-2,1},{1,-2,0}}.{x,y,z}
```

gibt die Matrix $\begin{pmatrix} 2 & -1 & 1 \\ 0 & -2 & 1 \\ 1 & -2 & 0 \end{pmatrix}$ ein und berechnet das Produkt mit dem (symbolischen) Vektor $(x, y, z)^t$.

```
characteristic polynomial{{2,-1,1},{0,-2,1},{1,-2,0}}
```

berechnet das charakteristische Polynom dieser Matrix (zeichnet es, und schreibt es in alternativen Formen).

```
jordan decomposition{{0,-1,1},{0,-2,1},{1,-2,0}}
```

berechnet euch die Jordanzerlegung von dieser Matrix (M); Ihr erhaltet die Matrizen S und J mit $M = S \cdot J \cdot S^{-1}$ wobei S die verallgemeinerten Eigenvektoren enthält und J auf der Diagonalen die Eigenwerte trägt & Jordanblockgestalt hat.

0.3 MATLAB (Desktop, auf den Uni Rechnern)

- Wenn ihr etwas "Skripten" möchtet, z.B. die Jordanzerlegung von mehreren bestimmten Matrizen für n=1,...,1000 haben wollt und die Resultate in einer CSV Datei
- Wenn ihr lineare Algebra über $\mathbb{C}$ oder $\mathbb{R}$ machen wollt
- Wenn ihr dreidimensionale Dinge (surface graphs) oder teilweise sogar vierdimensionale Dinge zeichnen wollt (hier braucht ihr "Slicing")
- Limitiert weil es kaum endliche Körper kann; Weil es ohne Erweiterung nicht symbolisch rechnen kann

Die gleichen Befehle in **Matlab** sehen so aus:

```
A=[2 -1 1; 0 -2 1; 1 -2 0];
```

(das Produkt mit einem symbolischen Vektor ist nicht Standard in MATLAB; Das Semikolon unterdrückt die Ausgabe. Lasst ihr es weg, kriegt ihr den Output angezeigt.)

```
charpoly(A)
```

(ohne ; erhaltet Ihr den Output der Funktion im Fenster abgedruckt)

```
[B,C]=jordan(A)
```

(wenn Ihr nur jordan(A) eingebt erhaltet ihr nur die Jordangestaltmatrix C, und nicht auch die veralg. Eigenvektoren B; so erhaltet Ihr beide Outputs der Funktion.)

0.4 Singular (Online, kostenlos, Falkos Favorit)

Webinterface: https://www.singular.uni-kl.de:8003/.

Singular ist relativ kompliziert aufgebaut. Allerdings könnt ihr damit auch Rechnungen über endlichen Körpern (oder beliebigen Ringen) durchführen. Um z.B. *Aufgabe 1 von Blatt 4* zu verifizieren geht ihr wie folgt vor:

```
ring A1 = 3,x,dp;
```

Definiert euren Grundkörper als $\mathbb{F}_3$. Dann ladet ihr die notwendigen Bibliotheken mit

```
LIB "all.lib";
```

(; nicht vergessen, in Singular kriegt ihr den Output so oder so irgendwie. Aber ohne ";" geht der Befehl nicht durch.) Die Matrix definiert ihr euch durch Zeilenweise aufschreiben:

```
matrix A[5][5]=2,2,0,0,0,1,1,0,2,0,2,0,2,1,0,2,0,0,0,0,1,2,2,0,2;
```

Dann könnt ihr z.B.

```
jordan(A);
```

benutzen um die Jordanstruktur zu erhalten. In diesem Fall erhaltet ihr

```
[1]:  _[1]=-1 _[2]=-1 [2]:  2,3 [3]:  1,1
```

(Eigenwerte sind -1 und -1; Die Jordannormalform hat zwei Jordanblöcke einen der Größe 2, einen der Größe 3; die Vielfachheiten der Jordanblöcke sind in [3] angegeben, nämlich 1 und 1.) Zum Bestimmen der Jordanbasis benutzt ihr (wie gesagt, Singular ist umständlich):

```
list l=jordanbasis(A);
print(l[1]);
```

Der Output ist dann die Jordanbasis

```
1,0, -1,0,0,
0,1, 0, 0,0,
0,-1,-1,1,0,
0,-1,-1,0,0,
0,1, 0, 0,-1
```

Weitere Befehle sind

```
inverse(A); oder charpoly(A);
```

zum Invertieren einer Matrix mit Einträgen in diesem endlichen Körper und berechnen des charakteristischen Polynoms.

Noch mehr findet ihr in singular linalg.lib Referenz (*linalg.lib*) und der Dokumentation der *matrix.lib*.

0.4.1 SageMath (Online oder Desktop)

- Benutze, falls die anderen Tools dein Problem nicht bearbeiten können
- Usability ist nicht im Fokus

Die kostenfreie Open Source Variante zu den kommerziellen Lösungen (MATLAB, Mathematica, Maple). Das heisst, es ist komplizierter/komplexer aufgebaut als MATLAB & Co. Dafür ist es auch extrem divers (es integriert z.B. alle Funktionen von **Singular**).

Kostenfreie Version, Desktop: www.sagemath.org
Kostenfreie Browser Version: sagecell.sagemath.org
(**Cloud Version:** cloud.sagemath.com)

SageMath kann alles, was die anderen Programme auch können. Hier ist ein Beispiel, welches ich so nicht auf Anhieb mit den anderen angehen kann. Für Gruppentheorie integriert **SageMath GAP** (das ist aber für euch egal). Ich nutze hier die Online Version SageCell und bestimme die Konjugationsklassen von einer Permutation (in Zykelschreibweise) in S_5:

```
G = SymmetricGroup(5)
g = G((1,2,3,4,5))
G.conjugacy\_lass(g)
```

Der Output ist:

```
Conjugacy class of cycle type [5] in Symmetric group of order 5!  as a permutation group
```

Das heisst soviel wie für vordefinierte Gruppen gibt es einen Standardoutput. Die Konjugationsklassen von S_n sind bekannt, alle Zykel des Typs 5 sind in einer Konjugationsklasse und das lässt sich auch nachschlagen. Wir können aber auch eine beliebige Gruppe definieren und in dieser eine Konjugationsklasse ausrechnen lassen. Ein Beispiel für eine Matrizengruppe über endlichem Körper ist in Refrenz-Conjugacy-Classes ... to be continued)

Ihr könnt auch überprüfen, ob eine Untergruppe Normalteiler ist. Hier in A_4:

```
A4 = AlternatingGroup(4)
r1 = A4("(1,2) (3,4)")
r2 = A4("(1,3) (2,4)")
r3 = A4("(1,4) (2,3)")
H = A4.subgroup([r1, r2, r3])
H.is_normal(A4)
```

(Output $=$ **True**)

Ein ganz anderes Feld das **SageMath** bedient, dass die anderen Programme nicht bedienen ist Differentialgeometrie. Wir können z.B. eine "S^3" (Eine dreidimensionale Sphere) definieren und dann Dinge tun wie Krümmung ausrechnen, Koordinaten ausgeben lassen, Tensorfelder visualieren,... Definition über:

```
S3 = Manifold(3, 'S^3', latex_name=r'\mathbb{S}^3', start_index=1)
print(S3)
```

und dann weiter wie in diesem Beispiel: SageManifolds.... to be continued.

0.4.2 Mathematica, Maple, R

Mathematica und Maple sind kommerzielle Software Tools die ebenfalls vieles Rechnen können... (to be continued)