Dokumentation von schule2e

von F. Borchers
Version 0.1.1 vom 5. Dezember 2023

Verfügbar unter `https://github.com/fborchers/schule2e/`

Inhalte des Kapitels

Für Ungeduldige

- Mit der Dokumentenklasse `exam2e` werden Klassenarbeiten gesetzt. Es baut auf `exam` auf, der Dokumentenklasse von P. Hirschhorn, und stellt auf die deutsche Sprache um.
- Der Dokumentation `schule2e.pdf` ist im Anhang ein Beispiel aus dem Alltag zur Ansicht beigefügt. Diese Klassenarbeit wurde aus `ka4-worked-example.tex` kompiliert.
- Das Stylesheet `mathe2e` lädt `amsmath` und passt an den Gebrauch in der Schule an. Umgebungen für lineare Gleichungsysteme und Vektoren werden bereit gestellt. Mit `siunitx` werden Einheiten gesetzt.
- `gess.sty` enthält einige spezifische Formatierungen für `exam2e`, die wir an unserer Schule verwenden.
- Das Stylesheet `exam2e.sty` dient der Kompatibilität, wenn man die Aufgaben von Klassenarbeiten in einem Skript sammeln will (z.B. Dokumentenklasse `article`).
- Die Pakete `schule2e` sind kein Fork des Projektes von P. Breitfeld. Das Paket ist unseres Wissens nicht mehr verfügbar. Der Name des Paketes ist diesem Andenken gewidmet.

Übersicht schule2e

Sie kennen sich bereits ein wenig mit LaTeX aus, haben vielleicht an der Uni mal damit gearbeitet; eine Anleitung haben Sie bereits gelesen, z.B. die klassische LaTeX2e-Kurzbeschreibung. Nun wollen Sie das Textsatzsystem als Lehrer/in verwenden.

Die Pakete, die hier als schule2e zusammengefasst sind, sollen startklar sein, minimal und doch umfassend für die täglichen Aufgaben in der Schule, ins. das Erstellen von Klassenarbeiten. Dazu sind die Pakete modular aufgebaut: Sie nehmen sich den Teil, den Sie gebrauchen können, und auf den Rest verzichten Sie oder probieren es später aus. Wenn Sie später Details ändern möchten oder Ihre eigenen Klassen bauen werden, umso besser.

Die Dokumentenklasse exam2e.cls

Als Grundbestandteil betrachten wir die Dokumentenklasse exam2e.cls. Die Sprache wird auf deutsch gestellt und die Bezeichnungen der Aufgaben und Unteraufgaben wird angepasst. Teilaufgaben werden ab 1.1 durchnummeriert, Unteraufgaben wie üblich mit (a), (b) usw. durchgezählt. Entscheidend ist, dass man *die* Struktur verwenden kann, die man für die eigenen Aufgaben benötigt. Die Klassenarbeit ka1 zeigt ein Minimalbeispiel auf, ka4 eine echte Klassenarbeit aus unserem Unterricht (sie ist dem PDF dieses Dokumentes angehängt, siehe unten).

Die Klasse exam2e ruft intern die Dokumentenklasse exam von Philip Hirschhorn auf, daher sei auf deren Dokumentation (englisch) verwiesen für alle Detailfragen. exam ist ein wohl gepflegtes, umfangreiches Projekt, ohne das exam2e in dieser Form nicht möglich wäre.

exam2e und das \aaa-Makro

Mit dem Makro \aaa haben Sie einen feinkörnigen Zugriff auf den Zähler subpart der exam-Dokumentenklasse. Das ist vor allem dann sinnvoll, wenn man innerhalb einer (oder mehrerer) Zeilen Mathematiksatz Unteraufgaben stellen möchte. In ka2 werden diese Funktionen vorgestellt.

Das exam2e-Stylesheet (exam2e.sty)

Als zweiten, unabhängigen Grundbaustein verstehen wir das Paket exam.sty . Es wird *nicht* zusammen mit der Klasse exam2e.cls verwendet, sondern dient der Kompatibilität mit dieser Klasse. Dazu werden die Umgebungen question , part, subpart und subsubpart definiert und erzeugen einen vergleichbaren Output wie in der exam documentclass.

Das Paket ist für die Verwendung mit der Dokumentenklasse article, scrarticle, book usw. gedacht und macht die exam2e-Makros dort verfügbar. Wenn Sie also Aufgaben und Unterlagen in einem LaTeX-Dokument sammeln, können Sie denselben Markup und dieselben Befehle verwenden wie für die Prüfungsaufgaben auch. Später können Sie dann die Aufgaben durch

Kopieren und Einfügen in Klassenarbeiten (documentclass exam2e) einfließen lassen, oder umgekehrt, Prüfungsaufgaben aus einer Klassenarbeit herauskopieren und in den Skripten archivieren. Es ist genau dieses Zusammenspiel, das einigen Mehrwert verspricht.

Das Beispiel unter `dokumentation/einskript` zeigt, wie eine solche Sammlung aussehen kann. Das Paket exam2e wird geladen, im Text befinden sich die Aufgaben in einer Umgebung `question` oder nach einem Befehl `\question`.

Unser Stylesheet `gess.sty`

In diesem Paket sind einige Funktionen zusammengefasst, die wir an unserer Schule nutzen. Insbesondere ist die Tabelle `klassenarbeitskopf` bei uns in Gebrauch; es ist höchstwahrscheinlich, dass Sie dieses abändern müssten, wenn Sie einen ähnlichen Kopf für Klassenarbeiten verwenden wollen. Ein kleinerer, allgemeiner Kopf für eine Klassenarbeit entsteht durch den Aufruf von `\klassenarbeitszeile`, die Syntax ist identisch.

Das Stylesheet `mathe2e.sty`

Das Paket mathe2e ist ebenfalls optional und kann sowohl in Klassenarbeiten als auch im Skript geladen werden. Es ist als Angebot gemeint, im besten Fall verwenden Sie natürlich Ihre eigenen Makros für den Mathematiksatz.

mathe2e lädt im Hintergrund amsmath und stellt damit alle Strukturen zur Verfügung. Wir ergänzen nur noch die, die für die Schulmathematik gebraucht werden: Zahlbereiche und Mengen, Relationen, lineare Gleichungssysteme (LGS) und die Vektorgeometrie. Zusätzlich wird siunitx geladen für die korrekte Darstellung der Einheiten und des richtigen Abstandes (ins. das Verhindern eines Zeilenumbruchs zwischen Zahl und Einheit). Damit können Einheiten ohne Zahlwerte, Prozente, Währungen und Kommazahlen entsprechend der deutschen Notation dargestellt werden.

Dank

Dieses Projekt begann irgendwann als Fork von P. Breitfelds Paket schule2e, geworden ist es eine Neufassung der wichtigsten Funktionen für die Nutzung in der Schule. Das Original schule2e ist leider nicht mehr verfügbar, sein letzter Beitrag zur Zeit der Niederschrift über zehn Jahre her. Es ist daher nicht anmaßend gemeint, den Namen schule2e weiterzuführen, sondern dankend.

Im „Maschinenraum“ von schule2e wird die Dokumentenklasse exam von P. Hirschhorn aufgerufen. exam ist eine mächtige und mit über 8000 Zeilen Quellcode umfangreiche Klasse, die die Struktur der Aufgaben und auch den Punktzähler bereitstellt und die seit 30 Jahren gewissenhaft gepflegt wird.

Die Dokumentenklasse `exam2e.cls`

Struktur der Aufgaben (`\question`)

Die Befehle zur Aufgabenerstellung folgen im wesentlichen der Vorgabe der Dokumentenklasse `exam`[1]. Fragen werden durch ein `\question` oder durch ein `\titledquestion` begonnen und abschließend durch eine Kommentarzeile beendet, um die Übersichtlichkeit zu bewahren. Nach dieser Initialisierung einer Aufgabe folgen der Text, die Abbildungen usw. Die anschließenden Arbeitsaufträge werden als Unteraufgaben mit `\subpart` aufgezählt:

(a) Unteraufgabe

(b) Unteraufgabe

```
\begin{subparts}
  \subpart Unteraufgabe
  \subpart Unteraufgabe
\end{subparts}
```

[1] P. Hirschhorn. „Package for Typesetting Exam Scripts“. URL: `https://ctan.org/pkg/exam`.

Unteraufgaben in einer Zeile mit `\aaa{}`

In manchen Situationen, v.a. in der Unter- und Mittelstufe, werden mehrere Unteraufgaben innerhalb einer Zeile gestellt:

(c) $\sqrt{2} \cdot \sqrt{18}$ (d) $\sqrt{10} \cdot \sqrt{3{,}6}$ (e) $\sqrt{125} : \sqrt{5}$

Das Paket `schule2e` bietet dazu einen feinkörnigen Zugriff auf den Zähler von `subpart` an – weswegen oben auch die Nummerierung der Unteraufgaben weiterläuft (c–e). Das Kommando `\aaa` erhöht den Zähler um eins, druckt das Label mit den Klammern, forciert aber keinen Zeilenumbruch. Die obige Zeile wurde erzeugt von:

```
\begin{align*}
  \aaa & \quad \sqrt{2}\cdot \sqrt{18}
& \aaa & \quad \sqrt{10}\cdot \sqrt{\num{3.6}}
& \aaa & \quad \sqrt{125} : \sqrt{5}
\end{align*}
```

Der Befehl `\aaa` steht im Mathematiksatz und im normalen Modus zur Verfügung. Man kann ihn also auch in Tabellen, Listen, Abbildungen und tabbings verwenden. Ähnlich dem Makro `\LaTeX{}` für LaTeX ist ggf. eine leere Gruppe `{}` mitzugeben.

Teil- und Unteraufgaben

Die Aufgaben können aber auch als Teilaufgaben geladen werden. Dieses Format bietet sich vor allem in der Oberstufe an, der Unterschied besteht in der Hierarchie und der Darstellung (als 1.1 durchnummeriert oder alphabetisch mit (a), (b) usw.).

Teilaufgaben und Unteraufgaben können dabei gleichzeitig zum Einsatz kommen bei langen Aufgaben wie zum Beispiel im Abitur.

Im Beispiel ka2-showcase ist die vollständige Gliederung in part, subpart, subsubpart und choice vorgeführt. Das Beispiel ka4-worked-example zeigt ein Beispiel aus dem Alltag (das kompilierte Ergebnis findet sich im Anhang dieser Dokumentation). Hier wird auch gezeigt, dass nicht immer alle Ebenen verwendet werden müssen, man kann z.B. auch gleich zu den Unterunteraufgaben springen, die mit kleinen römischen Zahlen abgezählt werden:

i. asdf
ii. asdf
iii. asdf

```
\begin{subsubparts}
  \subsubpart asdf
  \subsubpart asdf
  \subsubpart asdf
\end{subsubparts}
```

Referenzierungen mit \label und \ref

Die Counter der Teil- und Unteraufgaben sind so eingestellt, dass auf die Aufgaben LaTeX-üblich Bezug genommen werden kann. Mit \label wird der Referenzpunkt definiert, mit \ref darauf verlinkt. Beispielsweise kann auf die Unteraufgabe (a) hingewiesen werden, wenn diese das entsprechende (im Text nicht sichtbare) Label trägt:

(a) lorem ipsum

```
\begin{subparts}
  \subpart\label{sbp:unteraufg1} lorem ipsum
\end{subparts}
```

Wie immer bei \label wird die Kategorisierung mithilfe *eines* Doppelpunktes : (hier sbp:) empfohlen. In den Klassenarbeiten ka2-5 finden sich verschiedene Beispiele für diese Referenzierungen.

Ankreuzmöglichkeiten

Die unterste Ebene von Teilaufgaben, die exam anbietet, ist die der Ankreuzmöglichkeiten. Hier sind die Unteraufgaben nicht mehr nummeriert, sondern anhand von kleinen Boxen aufgelistet, die ggf. anzukreuzen sind:

☐ Ankreuzmöglichkeit 1

☐ Ankreuzmöglichkeit 2

☐ Ankreuzmöglichkeit 3

```
\begin{checkboxes}
  \choice Ankreuzmöglichkeit 1
  \choice Ankreuzmöglichkeit 2
  \choice Ankreuzmöglichkeit 3
\end{checkboxes}
```

Modifizieren der Auflistungen

In der Dokumentenklasse `exam` lassen sich die Auflistungen verändern. Im Abitur sollen beispielsweise die Unteraufgaben nicht mit (a), (b) usw. aufgeführt werden, sondern mit einer weiteren Nummerierung. Unser Beispiel `ka5-abiturstil` zeigt dieses Vorgehen, wenn die Unteraufgaben mit 1.1.1 fortgesetzt werden.

Spracheinstellung

Die Sprache wird von `exam2e` auf deutsch umgestellt. Falls das Dokument in englischer Sprache sein soll, kann mit

```
1 \usepackage[english]{babel}
```

einmalig auf Englisch umgestellt werden. Alle keywords von `exam` werden dazu übersetzt. Mehrsprachige Dokumente werden zurzeit nicht unterstützt, die Sprache kann nur einmalig zu Beginn des Dokumentes eingestellt werden.

mathe2e – eine Sammlung

Mathematische Notation mit `amsmath`

Mit `mathe2e` wird das Paket `amsmath`[2] geladen, das fast alle benötigten mathematischen Befehle bereitstellt. Wir fügen eine Übersicht der Zeichen für Zahlen, Relationen und Klammern an.

[2] American Mathematical Society. „User's Guide for the amsmath Package". URL: `https://ctan.org/pkg/amsmath`.

Zahlen	natürliche Zahlen	`\setN`	$\setN$
(je neu)	ganze Zahlen	`\setZ`	$\setZ$
	rationale Zahlen	`\setQ`	$\setQ$
	reelle Zahlen	`\setR`	$\setR$
	irrationale Zahlen	`\setR\setminus\setQ`	$\setR \setminus \setQ$
	irrationale Zahlen	`\setI` (selten)	$\setI$
	komplexe Zahlen	`\setC`	$\setC$
	Körper K	`\setK`	$\setK$
	Lösungsmenge	`\setL`	$\setL$
Gleichheit	definiere	`:=`	$:=$
	ungleich	`\neq}`	$\neq$
	identisch-gleich	`\equiv`	$\equiv$
	ungefähr gleich	`\approx`	$\approx$
	proportional zu	`\sim`	$\sim$
	kongruent zu	`\cong`	$\cong$
(neu)	erfordert gleich	`\mbeq`	$\mbeq$
(neu)	entspricht	`\hateq`	$\hateq$
Ungleichheit	kleiner als	`<`	$<$
	größer als	`>`	$>$
	kleiner gleich	`\leq`	$\leq$
	größer gleich	`\geq`	$\geq$
	viel kleiner als	`\ll`	$\ll$
	viel größer als	`\gg`	$\gg$
Relationen	Multiplikation	`\cdot`	$3 \cdot 12$
	Division	`:`	$12 : 3$
	Teilbarkeit	`\mid`	$3 \mid 12$
Klammern	Standard	`f(x)`	$f(x)$
	skaliert	`\left(...\right)`	$\left(1 + \frac{1}{2}\right)$
(neu)	Betrag	`\abs{...}`	$\abs{-2}, \abs{-\frac{1}{2}}$
(neu)	Norm	`\norm{...}`	$\norm{-2}, \norm{x}_2$
(neu)	Menge	`\set{...}`	$\set{a, b, c, d}$

Leider ist das Divisionszeichen (:) für LaTeX im deutschen Textsatz nicht perfekt eingestellt, man vergleiche etwa $3 : 4$ mit $3 \div 4$. Das `\colon` ist dagegen kein Relationszeichen, sondern ein Symbol der Zeichensetzung z.B. für $f\colon \mathbb{R} \to \mathbb{R}$, und führt damit bei der Division zu keinem guten Ergebnis, vergleiche $3\colon 4$.

Pfeile und beschriftete Pfeile

Pfeile	einfach	`\rightarrow`	$\rightarrow$
		`\leftarrow`	$\leftarrow$
	doppelt	`\Rightarrow`	$\Rightarrow$
		`\Leftarrow`	$\Leftarrow$
beschriftet	unskaliert	`\overset{.}{\rightarrow}`	$\overset{.}{\rightarrow}$
	skaliert	`\xrightarrow{...}`	$\xrightarrow{x\to\infty}$
(neu)	doppelt	`\xRightarrow{...}`	$\xRightarrow{\text{GTR}}$

Im einfachsten Fall kann man ein `overset` verwenden, um ein Symbol über den Pfeil zu drucken, dies funktioniert für links- wie rechtsgerichtete Pfeile. Wir zeigen noch ein Beispiel aus amsmath, den `\xrightarrow`, was den Pfeil horizontal skaliert, damit Platz für das Superskript bleibt:

$$ {}^{218}_{88}\mathrm{Ra} \xrightarrow{\alpha} {}^{214}_{86}\mathrm{Rn} + {}^{4}_{2}\mathrm{He} $$

In ähnlicher Weise wird das Makro `\xRightarrow` definiert, um daraus-folgt-Pfeile zu beschriften, z.B. bei der Verwendung des graphikfähigen Taschenrechners:

$$ \ldots \xRightarrow{\text{GTR}} \ldots $$

Einheiten mit `siunitx`

Wir laden das Paket `siunitx`[3] und setzen die Spracheinstellungen auf deutsch. Es folgt eine kurze Übersicht der Verwendung der Makros, im übrigen sei auf die Dokumentation des Paketes hingewiesen.

[3] J. Wright. „A Comprehensive (SI) Units Package“. URL: `https://ctan.org/pkg/siunitx`.

Zahlen und Mathematiksatz

Zahlen werden mit `\num{...}` eingefügt, wie z.B. $\pi \approx$ 3,141 59. Mit der Spracheinstellung deutsch wird ein Komma anstelle des englischen Punktes verwendet und auch die Abstände sind richtig nach dem Komma sowie nach je drei Ziffern. Der Befehl `\num` steht im normalen Text- als auch im Mathematikmodus zur Verfügung. Große oder kleine Zahlen werden im Makro mit `e` für die Zehnerpotenz gesetzt, so führt `\num{1.2e-3}` zu $1{,}2 \cdot 10^{-3}$.

Prozente und Promille werden mit `qty` geladen und fügen einen unbreakable space zwischen Zahl und Einheit: 5,1 % und 3 ‰. Winkel werden mit `\ang` gesetzt: 45°, komplexe Zahlen mit `complexnum`: 3,0 + 5,2i. Um konsistent zu sein, muss man dann immer mit `\complexnum{i}` (i) anstelle des einfachen `$i$` (i) arbeiten.

Größen mit Einheiten

Ähnlich den Prozenten werden auch die Einheiten mit `\qty{}{}` geladen. Als Beispiele seien aufgeführt 5 km, 5 $\frac{\text{km}}{\text{h}}$, 5 €. Das Setzen von Einheiten ist die

Stärke von siunitx, das Paket stellt alle benötigten Einheiten zur Verfügung. Wir geben als Beispiele an 1 N und 1 mA, die erzeugt wurden mit:

```
\qty{1}{\newton} und \qty{1}{\milli\ampere}
```

Als erstes Argument erwartet \qty einen numerischen Wert, als zweites die Einheit sowie ggf. den Vorsatz. Grundsätzlich kann man mit den ausgeschriebenen Namen der Einheiten arbeiten ohne ständig die Dokumentation des Pakets zu Rate ziehen zu müssen.

Die Vorsätze von Einheiten können für pädagogische Zwecke auch allein gedruckt werden, dazu wird die leere Einheit \noop bereitgestellt, die nichts druckt. Beispielsweise werden die Vorsätze k und G erzeugt mit:

```
\unit{\kilo\noop} und \unit{\giga\noop}
```

Gewählte Einstellungen von siunitx

Die Einstellungen von siunitx werden von dem Befehl \sisetup vorgenommen. Wir lassen das Paket siunitx folgendermaßen von mathe2e einladen:

```
\sisetup{% siunitx–Spracheinstellung deutsch:
  per-mode = fraction,
  locale = DE,
  list-final-separator = { und },
  list-pair-separator = { und },
  list-separator = {; },
  range-phrase = { bis }
}%
```

mathe2e: Zwei Makros für die Vektorrechnung

Anforderungen an geeignete Befehle für die Vektorrechnung

Wir fügen im folgenden an, welche Überlegungen zu der hier vorgestellten Implementierung geführt haben:

- Gesucht sind Befehle für Spaltenvektor (`\vector`) und für Zeilenvektor (`\coords`; für die Koordinaten von Punkten, getrennt durch je ein Pipe-Symbol „|“).
- Die Argumente (mindestens bis zu drei Zahlen) müssen in der gleichen Notation sowohl an `\vector` als auch an `\coords` übergeben werden können.
- Overloading der Definition, sodass mit zwei oder drei Argumenten bzw. Komponenten gearbeitet werden kann, z.B. `\vector(1,2)` für zweidimensionale Vektoren und `\vector(1,2,3)` für dreidimensionale. Entsprechend für `\coords`.
- Kompatibilität mit PSTricks `\psdot` und `\pstThreeDDot`, also eine Übergabe der Argumente in Klammern `()`, durch Komma `,` getrennt. So kann der Quellcode, insb. die Dokumentation von Darstellungen, mit copy-paste erarbeitet werden.
- Kompatibilität mit PSTricks `\rput` zur Beschriftung von geeigneten Grafiken.
- Der Spaltenvektor soll in einer Textzeile automatisch klein gedruckt werden (`\textstyle` gegenüber `\displaystyle`).
- Fehlermeldungen, die verständlich sind und auf die problematische Stelle stimmig hinweisen, insb. die entsprechende Zeile.

Eine mögliche Implementierung von \vector und \coords

Zum Einsatz des Befehls `\vector`:

$$\vec{v} = \begin{pmatrix} -3 \\ +4 \\ 5 \end{pmatrix} \quad \text{erzeugt mit} \quad \texttt{\textbackslash vector(-3,+4,5)}$$

Der Abstand des Vorzeichens ist automatisch richtig, genau wie bei inline-Vorzeichen (vgl. -3), im Gegensatz zum Abstand bei binären Operatoren (vgl. $3 - 4$). Die vertikalen Abstände bei überlappenden Brüchen können durch die Änderung von `arraystretch`– am besten in einer lokalen Gruppe – vergrößert werden, zum Beispiel wird

$$\begin{pmatrix} 3 \\ 4 \\ \frac{3}{2} \end{pmatrix} + \begin{pmatrix} 3 \\ \frac{3}{2} \\ \frac{4}{2} \end{pmatrix}$$

erzeugt von

```
\begingroup
\renewcommand{\arraystretch}{1.1}
\vector(3,4,\frac32) + \vector(3,\frac32,\frac42)...
\endgroup
```

Die inline-Version sieht entsprechend kleiner aus, wie hier der Vektor $\vec{v} = \binom{1}{2}{3}$. Man beachte, dass der Zeilenabstand entsprechend angepasst wird, was die Lesbarkeit des Textes, wie das gesamte Layout, beeinflusst. Vektoren mit nur zwei Komponenten sind gut zu setzen, $\vec{v} = \binom{1}{2}$, ohne große Beeinträchtigung.

Beispiele für Einsatz des Befehls `\coords`

Gegeben die folgenden Punkte:

$$P(1|3|5), \quad B\left(1\left|\tfrac{1}{2}\right|-3\right) \quad \text{und} \quad C(2|3)$$

die erzeugt wurden mit

```
P\coords(1,3,5), B\coords(1,\frac12,-3) und C\coords(2,3)
```

sowie inline $D(3|4)$ durch `$D\coords(3,4)$`. Die beiden Makros `\vector` und `\coords` sind darauf ausgelegt, auch mit mehr als drei Argumenten umzugehen, siehe die Erweiterungen unten.

Quellcode der Implementierung

Weil die Namen der Hilfmakros durch `@i` erweitert wurden, muss der Block innerhalb von `\makeatletter` stehen. So sind die Hilfsmakros beim Schreiben nicht zugänglich; nur `\vector` und `\coords` stehen dem Nutzer zur Verfügung.

```
\makeatletter
%% Definition von \vector zum Drucken von Spaltenvektoren :
\def\vector(#1){%
\mathchoice%
    {\pmatrix@i{\vector@i#1,,}}%
    {\pmatrix@ii{\vector@i#1,,}}%
    {}% scriptstyle too small,
    {}% ss style too small.
}% Ende Definition \vector. Hilfsfunktionen sind dabei :
\def\vector@i#1,{\if,#1,\else{#1}\cr\expandafter\vector@i\fi}
\def\pmatrix@i#1{\begin{pmatrix}#1\end{pmatrix}}
\def\pmatrix@ii#1{\left(\!\begin{smallmatrix}#1\end{smallmatrix}\!\
    right)}
%% Definition \coords zum Drucken von Zeilenvektoren :
\def\coords(#1,#2){\left(#1\coords@i#2,,\right)}
\def\coords@i#1,{\if,#1,\else|{#1}\expandafter\coords@i\fi}
\makeatother
\end{verbatim}%
```

Vergleich zu Standard-Makros

Ein Vergleich zur Umgebung `pmatrix`, die intern durch die Makros aufgerufen wird:

$$\begin{pmatrix}3\\4\\\frac{3}{2}\end{pmatrix}+\begin{pmatrix}4\\\frac{3}{2}\\\frac{4}{2}\end{pmatrix}$$

erzeugt mit

```
\begin{pmatrix}3\\4\\\frac32\end{pmatrix}
```

und ebenfalls vergößertem Wert für `\arraystretch`.

Mögliche Erweiterungen

Den Vektoren können als Argumente auch Unbekannte x mitgegeben werden (links). Ebenso sind Kommazahlen erlaubt:

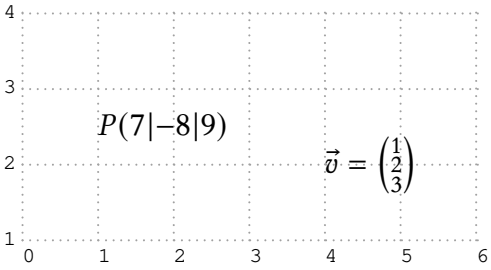
$$\begin{pmatrix}1\\2\\x\end{pmatrix}\quad\text{und}\quad\begin{pmatrix}3\\4\\5.6\end{pmatrix}\quad\text{bzw.}\quad\begin{pmatrix}7\\8\\9{,}0\end{pmatrix}\tag{*}$$

Bei der Kommasetzung läuft nur der anglizistische Punkt problemlos (Mitte). Ein Komma im Quellcode führt zu falschen Abständen (9, 0 verglichen mit 9,0), sowie zu einem handfesten Error, weil das Hilfsmakro `\vector@i` das Komma zur Trennung der einzelnen Komponenten einsetzt – wie gewünscht im Hinblick auf die Verwendung mit PSTricks. Eine Lösung ist der Einsatz des Pakets `siunitx` und dessen Funktion `\num{9.0}` für 9,0 (siehe Gl.(*), dritter Vektor dritte Komponente), dann ist auch der Abstand nach dem Komma korrekt dargestellt.

Darüber hinaus ist es auch möglich, die Makros einzusetzen, wenn mehr (oder weniger) Komponenten erforderlich sein sollten:

$$x^{\mu}=\begin{pmatrix}ct\\x\\y\\z\end{pmatrix}\quad\text{und}\quad x_{\mu}=(ct|-x|-y|-z)$$

Als Argumente zu PSTricks `\rput`-Befehl



Die Makros `\vector` und `\coords` können beide als Argument an PSTricks `\rput` übergeben werden. PSTricks `\rput` wird die Objekte im Modus „inline“ setzen.

mathe2e: Zwei Makros für lineare Gleichungssysteme

Gegeben sei ein lineares Gleichungssystem (LGS):

$$\left|\begin{array}{rcrcll} -2x_1 & + & 7x_2 & = & 6 & \text{(I)} \\ 4x_1 & - & 13x_2 & = & -1 & \text{(II)} \end{array}\right.$$

und das LGS als Matrix (man schreibt nur die Zahlen vor den Unbekannten):

$$\left(\begin{array}{cc|c} -2 & 7 & 6 \\ 4 & -13 & -1 \end{array}\right)$$

Man achtet dabei sorgfältig auf die Spalten, und setzt anstelle der Gleichheitszeichen einen Strich.

Interface der Umgebungen

Die Umgebungen `lgs` und `qmatrix` folgen derselben Struktur:

```
\begin{lgs}{m}{n}
   ...
\end{lgs}
```

```
\begin{qmatrix}{m}{n}
   ...
\end{qmatrix}
```

Die beiden notwendigen Argumente `m` und `n` enthalten die Anzahl der Spalten vor bzw. nach dem Gleichheitszeichen. Intern wird jeder Eintrag des LGS auf zwei Spalten aufgeteilt: eine mit dem Vorzeichen und eine mit dem Betrag. Für den obigen Eintrag $-2x_1$ wird also `& -&2x_1` geschrieben (s.u.). Der Vorteil ist, dass auf diese Weise die Vorzeichen korrekt ausgerichtet sind. Bei der Matrixschreibweise ist das nicht vorgesehen, die Einträge sind je zentriert.

Werden bei der Schreibweise als LGS Nummern für die Gleichungen gewünscht, werden diese manuell als `&(I)` usw. hinzugefügt. Man kann diese dann als Gl. (I) mit `Gl.~(I)` referenzieren; eine Verwendung von Labels ist an dieser Stelle nicht möglich. Im Mathematik-Modus ist ein `\rm` notwendig, um die Kursivschreibung zu verhindern, z.B. kann mit $2\cdot(\mathrm{I})+3\cdot(\mathrm{II})$ ein Lösungsweg dokumentiert werden.

Quellcode der Umgebungen `lgs` und `qmatrix`

```
\newenvironment{lgs}[2]{%
\left|% Füge nun den array ein:
\begin{array}{l@{}*{#1}{r@{\hspace{1pt}}r@{\;}}@{\;\,}>{\
   @ifnextchar=}c@{\;\;}*{#2}{r@{\hspace{1pt}}r@{\;}}@{\hspace{6mm
   }}>{\rm}l}}%
{\end{array}\right.}% Ende der Definition der Umgebung lgs.
%% Matrixform für LGS nach Format (cc...c|c):
\newenvironment{qmatrix}[2]{%
 \left(\begin{array}{@{}*{#1}{c}|*{#2}{c}@{}}%
}{%
 \end{array}\right)%
```

```
}% Ende der Definition der Umgebung qmatrix.
```

Quellcode der beiden Beispiele

```
\begin{lgs}{2}{1}
  & -&2x_1 & +& 7x_2 &==& &6 &(I) \\
  & &4x_1 & -&13x_2 &==& -&1 &(II)
\end{lgs}
```

für das LGS bzw. für die Matrix:

```
\begin{qmatrix}{2}{1}
  -2 & 7  & 6 \\
  4 & -13 & -1
\end{qmatrix}
```

Unser Paket gess.sty

Das Stylesheet `gess.sty` ist eine kleine Sammlung von Befehlen, die wir an unserer Schule verwenden. Wir erstellen damit den Kopf der Klassenarbeit, der die folgenden Informationen enthält: (1) Prüfungsart, (2) Fach, (3) Klasse und (4) Datum. Diese Klassenarbeitszeile hatte einer von uns in dieser Form jahrelang in Verwendung. Beispielsweise erzeugt man den folgenden Kopf mit:

```
\klassenarbeitszeile{Klassenarbeit 1}{Mathematik}{10c}{10.10.2023}
```

Vor- und Zuname: _____

Klassenarbeit 1, Mathematik 10c, am 10.10.2023

Aufgabe	1	2	3	4	Summe
Punkte	2	3	3	6	14

Abb. 1: Beispiel einer Kopfzeile für eine Klassenarbeit wie sie von dem Paket `gess.sty` erzeugt wird. Die Tabelle mit Punkten ist ein `\pointtable` aus der Dokumentenklasse `\exam`.

Einen etwas größeren Dokumentenkopf erhält man mit der Umgebung `klassenarbeitskopf`. Es ist nicht anzunehmen, dass dieser für Ihre Schule gut passt, weil er einige sehr spezielle Anpassungen enthält, die wir bei uns benötigen. Als Beispiel fügen wir an:

```
\begin{klassenarbeitskopf}{Klassenarbeit 1}{Mathematik}{10c
   }{10.10.2023}
Thema: \ldots \\
Zeit: 90 Minuten, Hilfsmittel: Taschenrechner \ldots
\end{klassenarbeitskopf}
```

Abb. 2: Ein Klassenarbeitskopf mit ausführlichen Angaben, wie er von dem Paket `gess.sty` erzeugt wird.

logo	Fach	Punkte	Note	Klassendurchschnitt	Sonstiger Leistungsstand
	Mathematik **Klassenarbeit 1**	von 14			

Vor- und Nachname	Klasse	Datum	Kenntnisnahme der Eltern
	10c	10.10.2023	

Thema: …
Zeit: 90 Minuten, Hilfsmittel: Taschenrechner …

Aufgabe	1	2	3	4	Summe
Punkte	2	3	3	6	14

Hinweis zur Summe der Punkte im Klassenarbeitskopf: Man könnte die Summe entfernen, indem man einen Patch einfügt, siehe tex.stackexchange.com. Das würde verhindern, dass sie Summe der Punkte doppelt im Klassenarbeitskopf erscheint.

Verwendung des `gess.sty` außerhalb von `exam`

Wenn man nur den Klassenarbeitskopf verwenden wollte ohne die Dokumentenklasse `exam`, dann wirft die interne Verwendung von gradetable bzw. pointtable einen Fehler auf. Man kann diesen umgehen, indem man den gradetable bzw. pointtable manuell erstellt und definiert:

```
\def\numpoints{14}% Anzahl Bewertungseinheiten <--
\def\pointtable[#1][#2]{% Argumente #1 und #2 werden ignoriert
\begin{tabular}{|l|*{5}{c|}}% Anzahl Fragen <--
\hline% -------------
Aufgabe&1&2&3&4&Summe\\% Aufgabennummern
\hline% -------------
Punkte &2&3&3&6&\numpoints\\% je Punkte
\hline% -------------
\end{tabular}}% Ende Definition \gradetable
```

Von diesem Vorgehen wird jedoch abgeraten.

logo	Fach	Punkte	Note	Klassendurchschnitt	Sonstiger Leistungsstand
	Mathematik				
	Klassenarbeit 1	von 51			

Vor- und Nachname	Klasse	Datum	Kenntnisnahme der Eltern
	10c	10.10.2023	

Thema: Vektorgeometrie
Zeit: 90 Minuten — Hilfsmittel: Taschenrechner, Formelsammlung.

Aufgabe	1	2	3	4	5	6	7	Summe
Punkte	4	6	4	7	7	19	4	51

Übersichtlichkeit, Darstellung und Rechtschreibung werden bewertet. Rechnungen müssen nachvollziehbar gestaltet werden. Achte bei allen Größen auf die richtige Einheit.

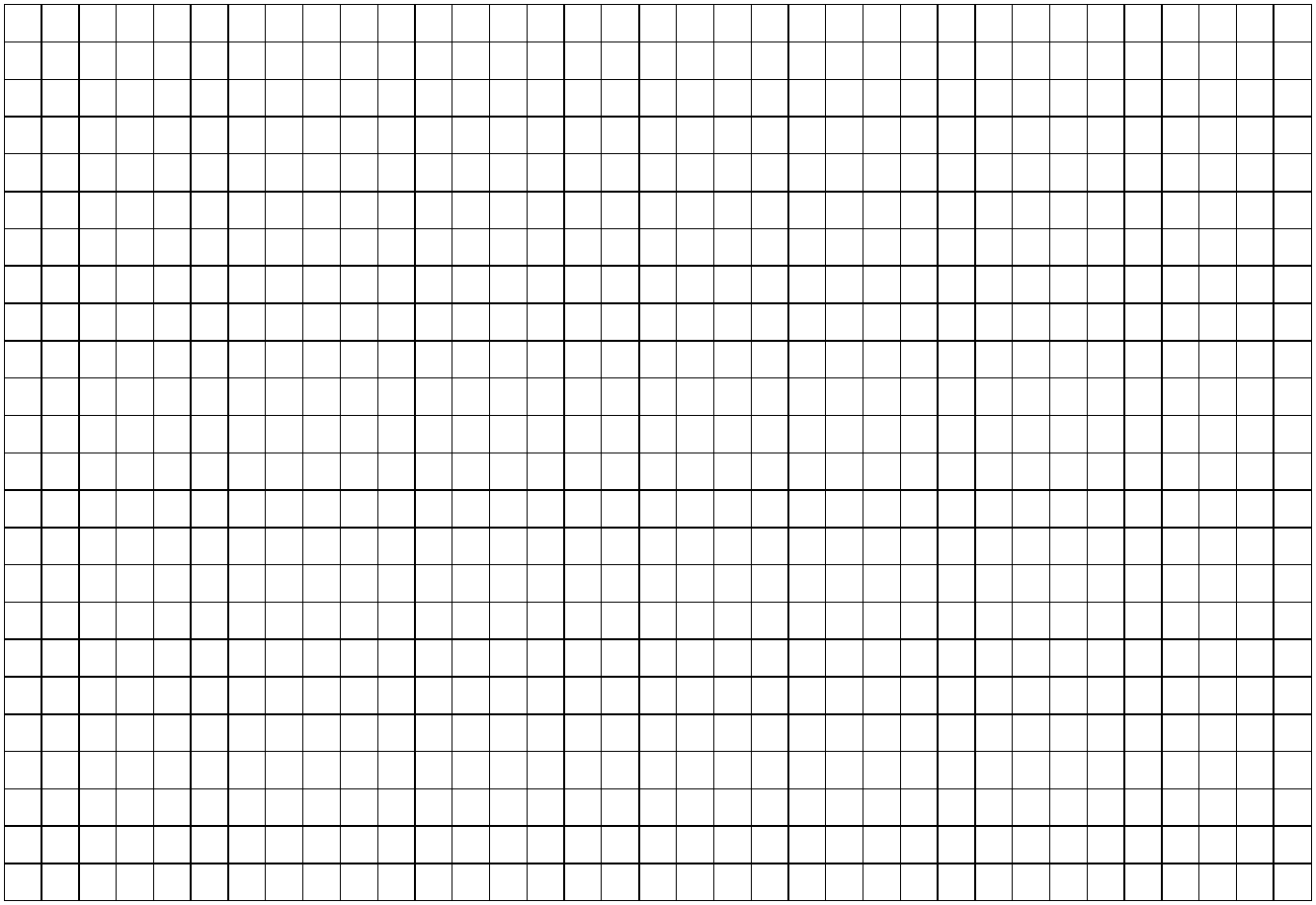
Teil 1: Keine Hilfsmittel zugelassen

Aufgabe 1

(4 Punkte)

Löse das folgende lineare Gleichungssystem:

$$\left|\begin{array}{rcl} 3x - 7y &=& 5 \\ -2x + 4y &=& -7 \end{array}\right.$$



Aufgabe 2 (6 Punkte)

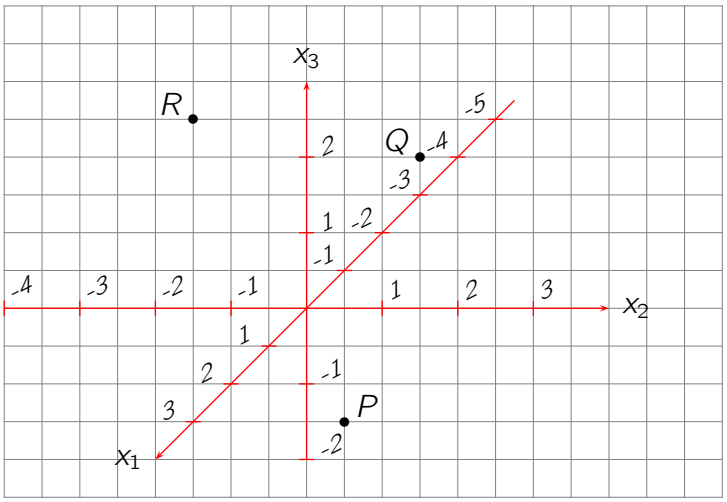
Im abgebildeten Koordinatensystem befinden sich

- der Punkt P in der x_1x_2-Ebene,
- der Punkt Q in der x_2x_3-Ebene,
- der Punkt R in der x_1x_3-Ebene.

(a) Bestimme die Koordinaten der drei Punkte P, Q und R.

(b) Gib den Ortsvektor $\overrightarrow{OP}$ an und zeichne ihn ein.

(c) Zeichne den Punkt $M(-5|2|-4)$ ein.



Aufgabe 3 (4 Punkte)

Widerlege die folgenden Aussagen:

(a) Wenn bei einem Vektor zwei Komponenten negativ sind, dann müssen bei einem anderen, parallelen Vektor diese Komponenten auch negativ sein.

(b) Wenn zwei Geraden einen gemeinsamen Spurpunkt haben, dann sind sie parallel oder identisch.

Teil 2: mit Taschenrechner und Formelsammlung

Aufgabe 4 (7 Punkte)

Es ist eine Pyramide mit quadratischer Grundfläche $ABCD$ und Spitze S gegeben.

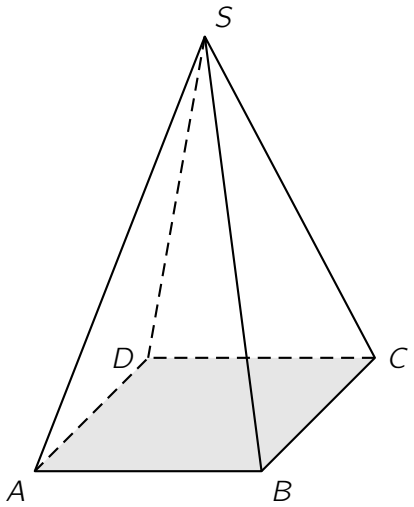


Abbildung 1: Nicht-maßstäbliche Abbildung einer Pyramide mit quadratischer Grundfläche

Von der Grundfläche sind die folgenden drei Punkte bekannt:

$$A(-5|-3|0), \quad C(12|4|0) \quad \text{und} \quad D(0|9|0).$$

Die Spitze S liegt bei $S(3{,}5|0{,}5|5)$.

(a) Bestimme die Koordinaten des Eckpunktes B.

(b) Berechne den Abstand des Mittelpunktes M_{BC} der Seite BC von der Spitze S der Pyramide.

Aufgabe 5 (7 Punkte)

Untersuche die Lagebeziehung der folgenden Geraden zueinander und bestimme gegebenenfalls den Schnittpunkt.

$$g: \vec{x} = \begin{pmatrix} \frac{2}{7} \\ 1 \\ 0 \end{pmatrix} + t \cdot \begin{pmatrix} -3 \\ -3 \\ -4 \end{pmatrix} \qquad \text{und} \quad h: \vec{x} = \begin{pmatrix} 0 \\ 9 \\ 1 \end{pmatrix} + s \cdot \begin{pmatrix} -5 \\ -7 \\ -7 \end{pmatrix}$$

Aufgabe 6 (19 Punkte)

Zwei Flugzeuge fliegen mit je konstanter Geschwindigkeit auf gradlinigen Flugbahnen. Die Position der Flugzeuge wird bezüglich eines Koordinatensystems mit der Längeneinheit 1 km angegeben, die x_3-Koordinate gibt die Flughöhe an. Um 8:00 Uhr ist das Flugzeug 1 im Punkt $P_1(-10|0|0)$ und das Flugzeug 2 im Punkt $P_2(-25|-30|8)$.

Wir betrachten im folgenden die Zeit t in min ab 8:00 Uhr: Nach vier Minuten hat das Flugzeug 1 die Position $Q_1(6|16|4)$ erreicht. Nach fünf Minuten hat das Flugzeug 2 die Position $Q_2(20|30|8)$ erreicht.

6.1 Erläutere, warum die Gleichung

$$g_1: \quad \overrightarrow{OX} = \begin{pmatrix} -10 \\ 0 \\ 0 \end{pmatrix} + t \cdot \begin{pmatrix} 4 \\ 4 \\ 1 \end{pmatrix} \qquad (*)$$

die Position des Flugzeugs 1 in Abhängigkeit von der Zeit angibt.

6.2 Ermittle analog zu Gleichung (*) in Teilaufgabe 6.1 eine Gleichung g_2, die die Position des Flugzeugs 2 in Abhängigkeit von der Zeit angibt.

6.3 Berechne die Geschwindigkeit des Flugzeugs 1 in $\frac{\text{km}}{\text{h}}$.

Der Luftraum, der von den Flugzeugen genutzt wird, wird von zwei verschiedenen Radarstationen überwacht. Die „Übergabe“ der Flugzeuge erfolgt, wenn die Flugzeuge die x_2x_3-Ebene durchfliegen.

6.4 Bestimme den Zeitpunkt und die Positions des Flugzeugs 1 bei der Übergabe.

6.5 Ermittle, zu welchem Zeitpunkt das Flugzeug 1 eine Flughöhe von 8 km erreicht und wie groß zu diesem Zeitpunkt der Abstand der beiden Flugzeuge ist.

Aufgabe 7 (4 Punkte)

Gegeben sei ein Vektor $\vec{v}$ mit einem noch unbestimmten Eintrag $a \in \mathbb{R}$:

$$\vec{v} = \begin{pmatrix} 17 \\ a \\ 8 \end{pmatrix}$$

Beurteile, ob es Werte a gibt, für die der Vektor $\vec{v}$ die Länge 18 hat.