easylinear — a LaTeX package for Linear Algebra

by Michael Kearney

2024-03-11 v1.2

1 About the easylinear package

The easylinear package was developed by a linear algebra student to make writing linear in LaTeX more easy. It was developed specifically for students of MATH0520 at Brown University, and some conventions in this package may be course-specific. It relies heavily on spalign by Joseph Rabinoff to define a limited set of efficient commands that save keystrokes, improve readability, and match the Octave syntax used in Sage cells more closely. This package is shared as is. The easylinear LaTeX package and associated template are *not* official course material. Use them at your own risk. The package has been thoroughly tested, but there is no guarantee of functionality. Always check your own work. See easylinear.sty for implementation specifics. Send feedback to michaelandrewkearney at gmail dot com.

2 How to set up easylinear

The easiest way to use easylinear is to make a copy of an Overleaf project that you can edit. Open the project, located at bit.ly/easylinear. Click "Menu" in the upper left corner and "Copy Project". (If you have already opened this template while signed into Overleaf, you can make a new copy directly from your Overleaf Projects page.) Choose a name for your new project (e.g. "Problem Set 1"). Your new project will contain main.tex, README.tex, and easylinear.sty. Write your document in main.tex. Compile and refer to README.tex for command usage. Do not edit easylinear.sty unless you know what you are doing. If you do mess up easylinear.sty, pull a fresh copy of the template. For more help with LaTeX see Overleaf's guide to Learn LaTeX in 30 minutes.

3 Updates to easylinear

I may update this package with more functionality. Existing commands will remain. Until indicated here, the Overleaf project is the most up-to-date version.

To get the latest version of `easylinear`, simply copy the project. Depending on maintenance capacity, future updates may be distributed over the fledgeling GitHub repository or CTAN.

4 Commands in `easylinear`

Use `easylinear` commands inside a math environment (i.e. inline `$\command$` or display `\[\command\]`). Pass an argument with curly braces like this: `\command{arg}`. Pass multiple arguments with multiple curly braces like this: `\command{arg1}{arg2}{arg3}`

5 Common commands

Input	Math	Input	Math
`x^2`	x^2	`\to`	$\to$
`\in`	$\in$	`\implies`	$\implies$
`\{`	$\{$	`\mathbb{R}`	$\mathbb{R}$
`\}`	$\}$	`\text{hello}`	hello

5.0.1 \paren{}

Use `\paren` to display something in appropriately-sized parentheses.

$$\left(\dfrac{x}{y}\right)$$

```
$\paren{\dfrac{x}{y}}$
```

6 Systems of Linear Equations

6.1 Linear system: \spalignsys

This command comes directly from the `spalign` package. It aligns the terms and operators of a system of equations and makes it easy to parse coefficients. `\spalignsys{}` takes one argument: a system of equations where terms are delimited by spaces or commas and equations are delimited by semicolons. For example:

$$\left\{\begin{array}{rcrcrcr} 2x_1 & - & 3x_2 & + & 12x_3 & = & 3 \\ -4x_1 & + & 0x_2 & + & 6x_3 & = & -10 \end{array}\right.$$

```
$\spalignsys{2x_1 - 3x_2 + 12x_3
= 3; -4x_1 + 0x_2 + 6x_3 =
-10}$
```

6.2 Vectors and Matrices

Use `\vct{...}` to make a column vector, `\mat{...}` to make a matrix, and `\amat{...}` to make an augmented matrix. Each command takes as an argument the matrix contents delimited like in Octave: delimit elements with spaces and/or commas and delimit rows with semicolons. Spaces collapse into each other and into commas, but commas do not collapse. For example:

$$\begin{bmatrix} 0 \\ 1 \\ 2 \end{bmatrix}$$

```
$\vct{0 1 2}$
```

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{bmatrix}$$

```
$\mat{0 1 2 3; 4 5 6 7}$
```

$$\begin{bmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{bmatrix}$$

```
$\mat{0,1 , 2 ,    3;4,5,6,7    ;}$
```

$$\left[\begin{array}{ccc|c} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \end{array}\right]$$

```
$\amat{0 1 2 3; 4 5 6 7}$
```

6.2.1 \dpr{}

Use `\dpr` to display the dot product of two vectors.

$$\begin{bmatrix} 2 \\ 4 \\ 5 \end{bmatrix} \bullet \begin{bmatrix} 3 \\ 8 \\ 0 \end{bmatrix}$$

```
$\dpr{2 4 5}{3, 8, 0}$
```

6.3 Row operations

There are three row operation symbol that describe row operations above a right arrow: `\rowop` represents addition/subtraction with optional scaling coefficients, `\rowscale` represents row scaling in-place, and `\rowswap` represents swapping two rows. Note that these commands do not perform row operations just like writing "3×4" does not actually compute a result. These operation symbols only represent the row operations you have done independently.

6.3.1 `\rowop{}{}{}{}{}`

`\rowop` takes five arguments: operated_coefficient, operated_row, operator, operand_coefficient, and operand_row. operated_row is the row to which operand_row is being added or subtracted. The result is stored in operated_row. The arguments are best understood by example:

$$\left[\begin{array}{ccc|c} 0 & 1 & 2 & 3 \\ 1 & 2 & 1 & -3 \end{array}\right] \xrightarrow{-3R_1+2R_2\to R_1} \left[\begin{array}{cc} 2 & 1 \\ 1 & 2 \end{array}\right.$$

```
\[\amat{0 1 2 3; 1 2 1 -3}
\rowop{-3}{1}{+}{2}{2}
\amat{2 1 -4 -15; 1 2 1 -3}\]
```

Only the row and operator arguments are mandatory. Coefficient arguments can be independently omitted (implying a coefficient of 1), but their curly brackets must remain. Any operator can be used. For example:

$$\left[\begin{array}{ccc|c} 0 & 1 & 2 & 3 \\ 1 & 2 & 1 & -3 \end{array}\right] \xrightarrow{R_1-R_2\to R_1} \left[\begin{array}{cc} -1 & -1 \\ 1 & 2 \end{array}\right.$$

```
\[\amat{0 1 2 3; 1 2 1 -3}
\rowop{}{1}{-}{}{2}
\amat{-1 -1 1 6; 1 2 1 -3}\]
```

6.3.2 `\rowscale{}{}`

`\rowscale` takes two arguments: operand and operated_row. operated_row is scaled by operand and stored back in operated_row.

$$\left[\begin{array}{ccc|c} 0 & 1 & 2 & 3 \\ 1 & 2 & 1 & -3 \end{array}\right] \xrightarrow{3R_1\to R_1} \left[\begin{array}{ccc|c} 0 & 3 & 6 & 9 \\ 1 & 2 & 1 & -3 \end{array}\right]$$

```
\[\amat{0 1 2 3; 1 2 1 -3}
\rowscale{3}{1}
\amat{0 3 6 9; 1 2 1 -3}\]
```

6.3.3 `\rowswap{}{}`

`\rowswap` takes two arguments: first_row and second_row. first_row and second_row are swapped.

$$\left[\begin{array}{ccc|c} 0 & 1 & 2 & 3 \\ 1 & 2 & 1 & -3 \end{array}\right] \xrightarrow{R_1\leftrightarrow R_2} \left[\begin{array}{ccc|c} 1 & 2 & 1 & -3 \\ 0 & 1 & 2 & 3 \end{array}\right]$$

```
\[\amat{0 1 2 3; 1 2 1 -3}
\rowswap{1}{2}
\amat{1 2 1 -3; 0 1 2 3}\]
```

6.4 Sets and solution sets

6.4.1 `\bracket{}` and `\set{}`

`\bracket{}` and `\set{}` surround an arbitrary argument with appropriately sized brackets. `\bracket{}` displays the contents as given, while `\set{}` will clean and comma-delimit the contents as much as possible.

$$\left\{ \text{Big brackets! } \begin{bmatrix} 2 & 4 \\ 3 & 2 \\ 7 & 6 \end{bmatrix} \right\}$$

```
$\bracket{\text{Big brackets! }\mat{2 4; 3 2; 7 6}}$
```

$$\left\{ 0, 1, \begin{bmatrix} 9 \\ 8 \end{bmatrix}, 3, \begin{bmatrix} 0 & 1 \\ 9 & 6 \end{bmatrix} \right\}$$

```
$\set{0 1, \vct{9 8},3; \mat{0 1; 9 6}}$
```

6.4.2 `\setwhere{}{}`

`\setwhere` adds a qualifying ‘such that’ statement to `\set` as a second argument. This ‘such that’ statement is not affected by the comma delimitation of `\set`.

$$\left\{ \begin{bmatrix} a \\ b \end{bmatrix}, \begin{bmatrix} c \\ d \end{bmatrix} \middle| \, a, b, c, d \in \mathbf{R} \right\}$$

```
$\setwhere{\vct{a b} \vct{c d}}{a,b,c,d \in \mathbf{R}}$
```

6.4.3 `\emptyset`

For inconsistent systems with no solution, the solution set is the empty set.

$\left[\begin{array}{cc|c} 1 & 0 & 1 \\ 0 & 0 & 1 \end{array}\right]$ has the solution set $\emptyset$.

```
$\amat{1 0 1; 0 0 1}$ has the solution set $\emptyset$.
```

6.4.4 `\unisolset{}`

Use `\unisolset{}` for consistent systems with a unique solution. It takes one argument: the terms of the solution vector delimited with spaces. For example:

$\left[\begin{array}{ccc|c} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 2 \\ 0 & 0 & 1 & 3 \end{array}\right]$ has the solution set $\left\{ \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \right\}$.

```
$\amat{1 0 0 1; 0 1 0 2; 0 0 1 3}$ has the solution set $\unisolset{1 2 3}$.
```

6.4.5 `\infsolset{}{}`

Use `\infsolset{}{}` for consistent systems with infinitely many solutions. The command takes two arguments. Delimit the terms of the solution vector with spaces in the first argument. Delimit the unbound variables with commas in the second argument. For example:

$\left[\begin{array}{ccc|c} 1 & -2 & 3 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}\right]$ sol'n set $\left\{ \begin{bmatrix} 2a-3b \\ a \\ b \end{bmatrix} \middle| a,b \in \mathbb{R} \right\}$.

```
1 $\amat{1 −2 3 0; 0 0 0 0; 0 0 0
  0}$ sol'n set $\infsolset{2a−3b a
  b}{a,b}$.
```

6.5 Putting it all together

This simple example shows how to combine `easylinear` commands. It may or may not be an example of good writing by this course's standards.

Problem 1. *Find the solution set of the linear system* $\begin{cases} 4x_1 + 2x_2 = 6 \\ 2x_1 + 0x_2 = 4 \end{cases}$.

From the linear system we derive the augmented matrix below. Using row operations, we transform it into RREF:

$$\left[\begin{array}{cc|c} 4 & 2 & 6 \\ 2 & 0 & 4 \end{array}\right] \xrightarrow{\frac{1}{2}R_2 \to R_2} \left[\begin{array}{cc|c} 4 & 2 & 6 \\ 1 & 0 & 2 \end{array}\right] \xrightarrow{R_1 \leftrightarrow R_2}$$

$$\left[\begin{array}{cc|c} 1 & 0 & 2 \\ 4 & 2 & 6 \end{array}\right] \xrightarrow{R_2 - 4R_1 \to R_2} \left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 2 & -2 \end{array}\right] \xrightarrow{\frac{1}{2}R_2 \to R_2}$$

$$\left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & -1 \end{array}\right]$$

Each variable is bound and there are no contradictions. There is one unique solution. This system of equations has the solution set $\left\{\begin{bmatrix} 2 \\ -1 \end{bmatrix}\right\}$.

```latex
\begin{problem}Find the solution set of the linear\\ system $\spalignsys{4x_1 + 2x_2 = 6; 2x_1 + 0x_2 = 4}$.
\end{problem}

From the linear system we derive the augmented matrix below. Using row operations, we transform it into RREF:
\begin{align*}
&\amat{4 2 6; 2 0 4}
\rowscale{\frac{1}{2}}{2}
\amat{4 2 6; 1 0 2}
\rowswap{1}{2}\\~\\
&\amat{1 0 2; 4 2 6}
\rowop{}{2}{-}{4}{1}
\amat{1 0 2; 0 2 -2}
\rowscale{\frac{1}{2}}{2}\\~\\
&\amat{1 0 2; 0 1 -1}
\end{align*}
Each variable is bound and there are no contradictions. There is one unique solution. This system of equations has the solution set $\unisolset{2 -1}$.
```

7 Vector Spaces

7.1 Commonly used commands

Input	Math	Input	Math
`\oplus`	$\oplus$	`\bigoplus`	$\bigoplus$
`\odot`	$\odot$	`\bigodot`	$\bigodot$

7.2 Spans and Linear Combinations

7.2.1 \vlist{}

Use `\vlist` to display a comma-separated list of vectors.

$$\begin{bmatrix}0\\5\\6\\3\end{bmatrix}, \begin{bmatrix}5\\4\\2\\7\end{bmatrix}, \begin{bmatrix}9\\9\\0\\4\end{bmatrix}$$

```
$\vlist{0 5 6 3; 5 4 2 7; 9 9 0 4}$
```

7.2.2 \vset{}

Use `\vset` to display a comma-separated set of vectors.

$$\left\{\begin{bmatrix}0\\5\\6\\3\end{bmatrix}, \begin{bmatrix}5\\4\\2\\7\end{bmatrix}, \begin{bmatrix}9\\9\\0\\4\end{bmatrix}\right\}$$

```
$\vset{0 5 6 3; 5 4 2 7; 9 9 0 4}$
```

7.2.3 \vspan{}

Use `\vspan` to display the span of a linear combination of a set of vectors.

$$\operatorname{span}\left\{\begin{bmatrix}0\\5\\6\\3\end{bmatrix}, \begin{bmatrix}5\\4\\2\\7\end{bmatrix}, \begin{bmatrix}9\\9\\0\\4\end{bmatrix}\right\}$$

```
$\vspan{0 5 6 3; 5 4 2 7; 9 9 0 4}$
```

7.2.4 \rspace{}

Use `\rspace` to display a Euclidean vector space of a certain dimension.

$\mathbb{R}^3$

```
$\rspace{3}$
```

8 Changelog

1.0 Published 2/14/23. First release. Added `\mat`, `\amat`, `\rowop`, `\rowscale`, `\rowswap`, `\unisolset`, and `\infsolset` commands.

1.1 Published 2/27/23. Added changelog. Added `\vct`, `\set`, `\setwhere`, `\rspan`, `\lincom`, `\vecset`, and `\vspan` commands. Added `\bracket` and `\commadelim` helper commands. Updated `\mat` and `\amat` to reference `spalign` commands directly. Updated `\infsolset` to use helper functions. Updated `{}` as default `spalign` delimiters. Updated README.

1.2 Published 3/11/24. Added `\dpr` and `\paren`. Added to github. Updated Overleaf template. Updated for Brown CS1420.

9 Acknowledgements

Thank you to Joseph Rabinoff, the developer of spalign, around which this package is a thin wrapper.
Thank you to Jordan Kostiuk, whose environment this package uses.
Thank you to Jordan Kostiuk and Sarah Griffith, in whose class this package was developed.
Thank you to KissMyArmpit, from whom this documentation styling was lifted.