asdia.sty Documentation

https://asdia.dev/projects/asdia.sty

Eytan Chong

April 2, 2025

Contents

1 Introduction

asdia.sty is a LaTeX style file for homework and assignments. It is inspired by Pascal Michaillat's latex-math and Evan Chen's evan.sty.

2 Packages

asdia.sty automatically loads the following packages and libraries:

- adjustbox
- amsmath
- amssymb
- amsthm
- bm
- booktabs
- caption
- chngcntr
- dsfont
- empheq
- enumitem
- etoolbox
- fancybox
- float
- hyperref
- mathtools
- mdframed
- multirow
- pgfplots
- systeme
- tabularx
- tasks
- thmtools
- tikz
- tkz-euclide
- xcolor
- xparse

2.1 Configurations

2.1.1 Hyperref

External links are configured to be RubineRed, internal links are RoyalBlue and citations are ForestGreen.

2.1.2 Tasks

Tasks are enumerated by lower-case alphabets:

(a) Hello (b) World (c) Foo (d) Bar

2.1.3 Xcolor

Three custom colours are introduced:

- `plotRed`
- `plotBlue`
- `plotGreen`

2.1.4 Tikz and Pgfplots

`asdia.sty` automatically loads the following `tikz` libraries:

- `angles`
- `arrows`
- `calc`
- `decorations.markings`
- `intersections`
- `patterns`
- `quotes`
- `shapes`

The following `pgfplot` libraries are also loaded:

- `polar`
- `fillbetween`

3 Brackets and Accents

3.1 Brackets

The following table lists all brackets provided by `asdia.sty`:

Description	Command	Example
Parentheses	`\bp`	$(\cdot)$
Square Brackets	`\bs`	$[\cdot]$
Braces	`\bc`	$\{\cdot\}$
Angle Brackets	`\ba`	$\langle\cdot\rangle$
Floor Function	`\floor`	$\lfloor\cdot\rfloor$
Ceiling Function	`\ceil`	$\lceil\cdot\rceil$
Modulus	`\abs`	$\lvert\cdot\rvert$
Norm	`\norm`	$\lVert\cdot\rVert$

Note that they all have the same syntax:

Syntax

`\foo{<argument>}`

- `{<argument>}` (mandatory): The object to surround with brackets.

Note also that these brackets scale automatically. This is particularly useful in display mode:

Examples

- `$\bp{\frac{1}{2}}$` $\longrightarrow$ Inline: $\bp{\frac{1}{2}}$, Display: $\displaystyle\bp{\frac{1}{2}}$.

3.2 Accents

The following table lists all accents provided by `asdia.sty`:

Description	Command	Example
Over-line	`\ol`	$\overline{x}$
Over-arrow	`\oa`	$\overrightarrow{x}$
Under-line	`\ul`	$\underline{x}$
Wide hat	`\wh`	$\widehat{x}$
Wide tilde	`\wt`	$\widetilde{x}$

Like the brackets, these accents share the following syntax:

Syntax

`\foo{<argument>}`

- `{<argument>}` (mandatory): The object to accent.

4 `\DefCmd` and its Variants

4.1 `\DefCmd`

One frequently used command is `\DefCmd`, which defines a mathematical operator with parentheses that scale automatically.

Syntax (`\DefCmd`)

`\DefCmd{<macro>}{<operator>}`

- `{<macro>}` (mandatory): The macro to define.
- `{<operator>}` (mandatory): The name of the mathematical operator.

To declare a new operator called "foo", we can invoke `\DefCmd{\foo}{foo}`. This creates the command `\foo`, which has the following syntax:

Syntax (`\foo`)

`\foo{<argument>}`

- `{<argument>}` (mandatory): The argument of foo.

For instance, we can now call `\foo{x}`. Note that if the argument is delimited with braces, it will be surrounded by parentheses in the output, as demonstrated below:

Examples (`\foo`)

- `$\foo{x}$` $\longrightarrow \mathrm{foo}(x)$.
- `$\foo x$` $\longrightarrow \mathrm{foo}\, x$.

4.2 `\DefCmdBc`

A minor variation of `\DefCmd` is `\DefCmdBc`. The only difference is that the brackets are now curly (`\bc` instead of `\bp`). The syntax is identical to that of `\DefCmd`.

4.3 \DefCmdPow

\DefCmdPow is yet another variant of \DefCmd. As its name suggests, it allows the user to write powers (exponents) after the operator.

The syntax of \DefCmdPow is completely identical to that of \DefCmd: to declare a new operator, we invoke \DefCmdPow{\foo}{foo}.

The resulting command, \foo, has the following syntax:

Syntax (\foo)

\foo[<power>]{<argument>}

- [<power>] (optional): The power of foo.
- {<argument>} (mandatory): The argument of foo.

Some example outputs are as follows:

Examples (\foo)

- $\foo[2]{x}$ $\longrightarrow$ $\text{foo}^2(x)$.
- $\foo[2] x$ $\longrightarrow$ $\text{foo}^2 x$.

If no exponent is passed, the output is completely identical to that of \DefCmd.

4.4 \DefCmdCond

The last variation of \DefCmd is \DefCmdCond. It is primarily used for probabilities, expectations and variations in statistics. There are two main differences between \DefCmdCond and \DefCmd:

- The brackets used are now square (\bp is replaced by \bs).
- There is an additional (optional) argument for \foo for conditionals.

The syntax is, once again, completely the same as \DefCmd: to define a new operator, we invoke \DefCmdCond{\foo}{foo}.

The syntax for \foo is as follows:

Syntax (\foo)

\foo{<argument>}{<condition>}

- {<argument>} (mandatory): The argument of foo.
- {<condition>} (optional): The event to condition upon.

Some example outputs are as follows:

Examples (\foo)

- $\foo{x}{y}$ $\longrightarrow$ $\text{foo}[x \mid y]$.
- $\foo{x}$ $\longrightarrow$ $\text{foo}[x]$.

5 Common Functions

5.1 Trigonometric Functions

The following trigonometric commands are implemented using `DefCmdPow`:

- `\sin`
- `\cos`
- `\tan`
- `\csc`
- `\sec`
- `\cot`
- `\arcsin`
- `\arccos`
- `\arctan`
- `\arccsc`
- `\arcsec`
- `\arccot`

Examples

- `$\sin[2]{x}$` $\longrightarrow \sin^2(x)$.
- `$\cos[2] x$` $\longrightarrow \cos^2 x$.
- `$\arccot x$` $\longrightarrow \operatorname{arccot} x$.

5.2 Exponential and Logarithmic Functions

The `\exp` and `\ln` commands are implemented using `\DefCmd`.

Examples

- `$\exp{x}$` $\longrightarrow \exp(x)$.
- `$\ln x$` $\longrightarrow \ln x$.

6 Functions and Graphs

The inverse of function is provided by `\inv`, while the domain and range of a function is provided by `\dom` and `\ran` respectively. They share the following syntax:

Syntax

`\foo{<function>}`

- `{<function>}` (mandatory): The function to take the inverse/range/domain of.

Examples

- `$\inv f$` $\longrightarrow f^{-1}$.
- `$\dom f$` $\longrightarrow D_f$.
- `$\ran f$` $\longrightarrow R_f$.

7 Linear Algebra

7.1 Boldface Vectors and Matrices

Boldface vectors and matrices are provided by `\vec` and `\mat` respectively.

Examples

- `$\vec{v}$` $\longrightarrow \mathbf{v}$.
- `$\mat{M}$` $\longrightarrow \mathbf{M}$.

Most of the time, the delimiters can be omitted.

7.2 Column Vectors

2D, 3D and 4D column vectors are provided by `\cvecii`, `\cveciii` and `\cveciv` respectively. Obviously, they take in the same number of parameters as dimensions.

Examples

- `$\cvecii{1}{0}$` $\longrightarrow \begin{pmatrix} 1 \\ 0 \end{pmatrix}$.
- `$\cveciv{a}{b}{c}{d}$` $\longrightarrow \begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix}$.

Their inline counterparts are provided by `\cveciix`, `\cveciiix` and `\cvecivx` respectively.

Examples

- `$\cveciix{1}{0}$` $\longrightarrow (1,\, 0)^{\mathsf{T}}$.
- `$\cvecivx{a}{b}{c}{d}$` $\longrightarrow (a,\, b,\, c,\, d)^{\mathsf{T}}$.

7.3 Transpose

The transpose of a vector/matrix is provided by `\trp`. It does *not* take in any argument; it is a decorative command.

Example

- `$\vec v \trp$` $\longrightarrow \mathbf{v}^{\mathsf{T}}$.

7.4 Dot and Cross Product

The dot and cross product symbols are provided by `\dotp` and `\crossp` respectively.

Examples

- `$\vec u \dotp \vec v$` $\longrightarrow \mathbf{u} \cdot \mathbf{v}$.
- `$\vec u \crossp \vec v$` $\longrightarrow \mathbf{u} \times \mathbf{v}$.

7.5 Matrix Operations

The following matrix operations are implemented using `\DefCmd`.

- `\tr`
- `\det`
- `\Dim`
- `\Ker`
- `\Nullity`
- `\Range`
- `\Rank`
- `\Col`

Examples

- `$\tr{\mat A}$` $\longrightarrow \operatorname{tr}(\mathbf{A})$.
- `$\Dim \mat A$` $\longrightarrow \dim \mathbf{A}$.

The odd-one-out is the `\Span` operator, which is implemented using `\DefCmdBc`.

Examples (`\Span`)

- `$\Span{\vec u, \vec v}$` $\longrightarrow \operatorname{span}\{\mathbf{u}, \mathbf{v}\}$.
- `$\Span S$` $\longrightarrow \operatorname{span} S$.

8 Complex Numbers

8.1 Imaginary Unit

`\i` produces the symbol for the imaginary unit, i.

Example (`\i`)

- `$\i^2 = -1$` $\longrightarrow \mathrm{i}^2 = -1$.

8.2 Complex Conjugate

`\conj` produces the symbol for the complex conjugate.

Example (`\conj`)

- `$z \conj$` $\longrightarrow z^*$.

8.3 Real and Imaginary Parts, and Argument

The real and imaginary part commands are provided by `\Re` and `\Im` respectively. The argument (angle) command is provided by `\arg`. All three functions are implemented using `\DefCmd`.

Examples

- `$\Re{z}$` $\longrightarrow \operatorname{Re}(z)$.
- `$\Im z$` $\longrightarrow \operatorname{Im} z$.
- `$\arg{z}$` $\longrightarrow \arg(z)$.

9 Calculus

9.1 Differential

The differential symbol is provided by `\d`.

Syntax (\d)

`\d{<x>}`

- `{<x>}` (mandatory): The variable to take the differential of.

Example (\d)

- `$\d{x}$` $\longrightarrow$ $\mathrm{d}x$.

Most of the time, the delimiters are not needed.

9.2 Derivatives

The total and partial derivatives are provided by `\der` and `\pder`. Both commands share the following syntax:

Syntax

`\foo{<y>}{<x>}`

- `{<y>}` (mandatory): The function to differentiate.
- `{<x>}` (mandatory): The variable we are differentiating with respect to.

Examples

- `$\der{y}{x}$` $\longrightarrow$ $\frac{\mathrm{d}y}{\mathrm{d}x}$.
- `$\pder{z}{t}$` $\longrightarrow$ $\frac{\partial z}{\partial t}$.

Their inline equivalents are provided by `\derx` and `\pderx` respectively, with exactly the same syntax.

Examples

- `$\derx{y}{x}$` $\longrightarrow$ $\mathrm{d}y/\mathrm{d}x$.
- `$\pderx{z}{t}$` $\longrightarrow$ $\partial z/\partial t$.

9.3 Evaluations

The command `\evalder` evaluates a derivative at a particular point.

Syntax (\evalder)

`\evalder{<derivative>}{<point>}`

- `{<derivative>}` (mandatory): The function/derivative to evaluate.
- `{<point>}` (mandatory): The point to evaluate at.

Example (\evalder)

- `$\evalder{\der{y}{x}}{x = 2}$` $\longrightarrow$ $\left.\frac{\mathrm{d}y}{\mathrm{d}x}\right|_{x=2}$.

Similarly, the command `\evalint` evaluates a primitive over an interval.

Syntax (\evalint)

`\evalint{<primitive>}{<lower bound>}{<upper bound>}`

- `{<primitive>}` (mandatory): The primitive to evaluate.
- `{<lower bound>}` (mandatory): The lower bound to evaluate the primitive at.
- `{<upper bound>}` (mandatory): The upper bound to evaluate the primitive at.

Example (\evalint)

- `$\evalint{x}{x = 0}{1}$` $\longrightarrow [x]_{x=0}^{1}$.

9.4 Integrating Factor

The integrating factor symbol is provided by `\IF`.

Example (\IF)

- `$\IF$` $\longrightarrow$ I. F..

10 Statistics

10.1 Permutations and Combinations

The permutation and combination commands are provided by `\perm` and `\comb` respectively. The two functions share the following syntax:

Syntax

`\foo{<n>}{<r>}`

- `{<n>}` (mandatory): Total number of objects.
- `{<r>}` (mandatory): Number objects to permute/choose.

Examples

- `$\perm{10}{5}$` $\longrightarrow {}^{10}P_5$.
- `$\comb{10}{5}$` $\longrightarrow {}^{10}C_5$.

10.2 Probability, Expectation and Variance

The probability, expectation and variance functions are provided by `\P`, `\E` and `\Var` respectively. They are implemented using `\DefCmdCond`.

Examples

- `$\P{X = 1}$` $\longrightarrow \mathbb{P}[X = 1]$.
- `$\E{X = 1}{Y = 1}$` $\longrightarrow \mathbb{E}[X = 1 \mid Y = 1]$.

10.3 Distributions

10.3.1 Binomial Distribution

The binomial distribution is provided by `\Binom`.

Syntax (`\Binom`)

`\Binom{<n>}{<p>}`

- `{<n>}` (mandatory): Number of trials.
- `{<p>}` (mandatory): Probability of success.

Example (`\Binom`)

- `$\Binom{10}{0.5}$` $\longrightarrow \mathrm{B}(10, 0.5)$.

10.3.2 Poisson Distribution

The Poisson distribution is provided by `\Po`.

Syntax (`\Po`)

`\Po{<lambda>}`

- `{<lambda>}` (mandatory): The mean rate of the Poisson process.

Example (`\Po`)

- `$\Po{2}$` $\longrightarrow \mathrm{Po}(2)$.

10.3.3 Geometric Distribution

The geometric distribution is provided by `\Geo`.

Syntax (`\Geo`)

`\Geo{<p>}`

- `{<p>}` (mandatory): Probability of success.

Example (`\Geo`)

- `$\Geo{0.5}$` $\longrightarrow \mathrm{Geo}(0.5)$.

10.3.4 Uniform Distribution

The uniform distribution is provided by `\Uni`.

Syntax (`\Uni`)

`\Uni{<a>}{<b>}`

- `{<a>}` (mandatory): The lower bound.
- `{<b>}` (mandatory): The upper bound.

Example (\Uni)

- `$\Uni{0}{1}$` ⟶ $\mathrm{U}(0,1)$.

10.3.5 Exponential Distribution

The exponential distribution is provided by `\Exp`.

Syntax (\Exp)

`\Exp{<lambda>}`

- `{<lambda>}` (mandatory): The mean rate of the Poisson process.

Example (\Exp)

- `$\Exp{2}$` ⟶ $\mathrm{Exp}(2)$.

10.3.6 Normal Distribution

The normal distribution is provided by `\Normal`.

Syntax (\Normal)

`\Normal{<mean>}{<variance>}`

- `{<mean>}` (mandatory): The mean.
- `{<variance>}` (mandatory): The variance.

Example (\Normal)

- `$\Normal{0}{1}$` ⟶ $\mathrm{N}(0,1)$.

10.3.7 Student T Distribution

The Student T distribution is provided by `\StudentT`.

Syntax (\StudentT)

`\StudentT{<df>}`

- `{<df>}` (mandatory): Degrees of freedom.

Example (\StudentT)

- `$\StudentT{10}$` ⟶ $\mathrm{t}(10)$.

10.4 Hypothesis Testing

The null hypothesis and alternative hypothesis symbols are provided by `\nullhyp` and `\althyp` respectively.

Examples

- `$\nullhyp$` ⟶ H_0.
- `$\althyp$` ⟶ H_1.

11 Number Theory

11.1 LCM and GCD

The lowest common multiple (LCM) and greatest common divisor (GCD) of a set of numbers is provided by `\lcm` and `\gcd` respectively. Both commands are implemented using `\DefCmd`.

Examples

- `$\lcm{4, 5}$` ⟶ $\lcm{4, 5}$.
- `$\gcd{6, 7}$` ⟶ $\gcd{6, 7}$.

11.2 Legendre Symbol

The Legendre symbol is provided by `\legendre`.

Syntax

`\legendre{<a>}{<p>}`

- `{<a>}` (mandatory): An integer.
- `{<p>}` (mandatory): An odd prime.

Example

- `$\legendre{4}{7}$` ⟶ $\legendre{4}{7}$.

12 Miscellaneous

12.1 Euler's Number

Euler's number is provided by `\e`.

Example (`\e`)

- `$\e$` ⟶ $\e$.

12.2 Extrema

The following commands are implemented using `\DefCmdBc`:

- `\max`
- `\sup`
- `\argmax`
- `\min`
- `\inf`
- `\argmin`

Examples

- `$\max{1, 2}$` ⟶ $\max{1, 2}$.
- `$\argmax \sin x$` ⟶ $\argmax \sin x$.

12.3 Less Common Functions

The following functions are implemented using `\DefCmd`.

- `\sgn` - `\bigO` - `\smallO`

Examples

- `$\sgn{1}$` ⟶ $\sgn{1}$.
- `$\bigO{x}$` ⟶ $\bigO{x}$.
- `$\smallO{x}$` ⟶ $\smallO{x}$.

Another less common function is the indicator function, which is provided by `\ind`. It is implemented by `\DefCmdBc`.

Example (`\ind`)

- `$\ind{0<x<1}$` ⟶ $\ind{0<x<1}$.

12.4 Geometry

12.4.1 Degrees

The degree symbol is provided by `\deg`.

Example (`\deg`)

- `$180\deg$` ⟶ $180\deg$.

12.4.2 Measures

The following commands are implemented by `\DefCmd`:

- `\length` - `\area` - `\volume`

Examples

- `$\area \triangle ABC$` ⟶ $\area \triangle ABC$.
- `$\volume{ABCD}$` ⟶ $\volume{ABCD}$.

Units are provided by `\units`.

Syntax (`\units`)

`\units[<dimension>]`

- `[<dimension>]` (optional): The dimension of "units".

Examples (`\units`)

- `$10 \units$` ⟶ $10 \units$.
- `$10 \units[2]$` ⟶ $10 \units[2]$.

12.5 Text in Math Mode

12.5.1 Logical Connectives

`\tand`, `\tor` and `\ow` are used to print "and", "or" and "otherwise" in display equations. Spaces are automatically added before and after the words.

Examples

- `$P \tand Q$` ⟶ P and Q.

12.5.2 Precision

`\tosf` and `\todp` are used to indicate the precision of a value (significant figures and decimal places). Both have the following syntax:

Syntax

`\foo{<precision>}`

- `{<precision>}` (mandatory): The number of significant figures/decimal places the value is rounded off to.

Examples

- `$1.23 \tosf{3}$` ⟶ 1.23 (3 s.f.).
- `$1.23 \todp{2}$` ⟶ 1.23 (2 d.p.).

12.6 Cases and Subcases

`\case` and `\subcase` are used to label cases and subcases respectively. The two commands share the following syntax:

Syntax

`\foo{<label>}[<statement>]`

- `{<label>}` (mandatory): The case's number.
- `[<statement>]` (optional): The statement that case is considering.

Examples

- `\case{1}[$x = 1$]` ⟶ *Case 1*: $x = 1$.
- `\case{2}` ⟶ *Case 2*.

Note that a period is automatically added after the case/subcase command.

12.7 Section Sign

The section sign (§) is provided by `\SS`. Note that this command works in both text and math mode.

12.8 Letters

12.8.1 Greek Letters

`asdia.sty` provides shortcuts for most Greek letters: only omicron, pi (π) and tau (τ) do not have their own shortcuts.

Letter	Command	Letter	Command	Letter	Command
α	\a	β	\b	γ	\g
Γ	\G	δ	\de	Δ	\D
ϵ	\ep	ε	\ve	ζ	\z
η	\h	θ	\t	ϑ	\vt
Θ	\T	ι	\io	κ	\k
$\varkappa$	\vk	λ	\l	Λ	\L
μ	\m	ν	\n	ξ	\x
Ξ	\X	ρ	\r	ϱ	\vr
σ	\s	ς	\vs	Σ	\S
υ	\u	Υ	\U	ϕ	\f
φ	\vf	Φ	\F	χ	\c
ψ	\p	ω	\o	Ω	\O

12.8.2 Blackboard Letters

`asdia.sty` provides shortcuts for a select few blackboard letters:

Letter	Command	Letter	Command	Letter	Command
$\mathbb{C}$	\CC	$\mathbb{R}$	\RR	$\mathbb{Z}$	\ZZ
$\mathbb{Q}$	\QQ	$\mathbb{N}$	\NN	$\mathbb{F}$	\FF

12.8.3 Calligraphic Letters

`asdia.sty` provides shortcuts for all calligraphic letters:

Letter	Command	Letter	Command	Letter	Command
$\mathcal{A}$	\Ac	$\mathcal{B}$	\Bc	$\mathcal{C}$	\Cc
$\mathcal{D}$	\Dc	$\mathcal{E}$	\Ec	$\mathcal{F}$	\Fc
$\mathcal{G}$	\Gc	$\mathcal{H}$	\Hc	$\mathcal{I}$	\Ic
$\mathcal{J}$	\Jc	$\mathcal{K}$	\Kc	$\mathcal{L}$	\Lc
$\mathcal{M}$	\Mc	$\mathcal{N}$	\Nc	$\mathcal{O}$	\Oc
$\mathcal{P}$	\Pc	$\mathcal{Q}$	\Qc	$\mathcal{R}$	\Rc
$\mathcal{S}$	\Sc	$\mathcal{T}$	\Tc	$\mathcal{U}$	\Uc
$\mathcal{V}$	\Vc	$\mathcal{W}$	\Wc	$\mathcal{X}$	\Xc
$\mathcal{Y}$	\Yc	$\mathcal{Z}$	\Zc		