Template for a nice LaTeX document

NAME1, NAME2

October 20, 2024

Institute for Computational Physics, University of Stuttgart

Contents

1 Math

The package `amsmath` is available for typesetting math. Execute `texdoc amsmath` in your terminal for additional information.

Here is an equation typeset using commands from `amsmath`.

$$\alpha = \begin{pmatrix} D_1 t & -a_{12} t_2 & \dots & -a_{1n} t_n \\ -a_{21} t_1 & D_2 t & \dots & -a_{2n} t_n \\ \dots & \dots & \dots & \dots \\ -a_{n1} t_1 & -a_{n2} t_2 & \dots & D_n t \end{pmatrix} \tag{1}$$

$$\beta = \pi \tag{2}$$

Furthermore don't use `eqnarray` and use `align` instead! (If you don't know what `eqnarray` is, nevermind)

1.1 Units

Typesetting units by hand is extremely annoying because you have to take care of spacing and font style, for example

```
100 \; \mathrm{m} \text{ or } 100 \; \mathrm{\frac{m}{s}}
```

to produce

$$100 \; \mathrm{m} \text{ or } 100 \; \frac{\mathrm{m}}{\mathrm{s}}$$

That's why we use the package `siunitx` See:

```
\SI{100}{\meter} \text{ or } \SI{100}{\meter\per\second}
```

to produce 100 m or $100 \, \frac{\mathrm{m}}{\mathrm{s}}$

2 Code

You can include code using `lstlisting`

```
class HarmonicPotential():
  def __init__(self, k):
    self.k = k
  def calc_energy(self, x):
    return 0.5*self.k*x**2
  def calc_force(self, x):
    return -self.k*x

```