Contents

Listings

This is the new `mandi` package (v3.0.0alpha dated 2020/11/26), which creates a habitat specifically for introductory physics students. It provides consistent notation conventions that align with ISO 80000-2 recommendations. This document is a demonstration for the new package and should not be taken as official documentation. Features are not necessarily final and are still subject to change.

1 Fonts

1.1 Text Mode Fonts

1.1.1 Default Text Font Parameters

To begin with, let's look at the fonts we have at our disposal. Let's look at the default text mode font parameters first.

`\encodingdefault`	TU
`\familydefault`	lmr
`\seriesdefault`	m
`\shapedefault`	n
`\rmdefault`	lmr
`\sfdefault`	lmss
`\ttdefault`	lmtt
`\bfdefault`	b
`\updefault`	up
`\itdefault`	it
`\mddefault`	m
`\sldefault`	sl
`\scdefault`	sc

1.1.2 Text Mode Font Commands

Now let's look at the text mode commands for changing fonts and their accompanying results. Note that these commands require braced arguments and work only within the scope of the braces.

Text Mode Commands
The default normal text is `\textnormal{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get roman letters, use `\textrm{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get san serif letters, use `\textsf{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get typewriter letters, use `\texttt{...}`.
`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`
To get medium letters, use `\textmd{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get boldface letters, use `\textbf{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Text up `\textup{...}`
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get italic letters, use `\textit{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get slanted letters, use `\textsl{...}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get small capital letters, use `\textsc{...}`.
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

Some of those text mode commands may seem redundant to you, and there are some you will never need to use, so don't worry if they look confusing. Remember that `\textnormal{...}` is the default and all you have to do to get it is start typing; no command is necessary unless you have changed something (and that's usually difficult to do). These text font commands can be used together if you remember to nest the braces correctly. Look at the following table.

Combined Text Mode Commands
To get boldface san serif letters, use `\textbf{\textsf{...}}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get boldface italic letters, use `\textbf{\textit{...}}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
To get boldface slanted letters, use `\textbf{\textsl{...}}`.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

1.1.3 Emphasizing Words in Text Mode

For semantic reasons, it's important to use `\emph{...}` when you want to emphasize a word. Use `\emph{vector}` to emphasize the word *vector* or `\emph{momentum}` to emphasize the word *momentum*. In most documents, `\emph{...}` defaults to italics, but it may default to something else when using a different document class. Let the document class do its work and you will have fewer things to remember.

1.1.4 Text Mode Font Switches

Next, let's look as the various text mode switches you can use. They are called *switches* rather than *commands* because they take effect immediately and stay in effect until you explicitly turn them off or activate another one, much like a light switch. As so, they do not take braced arguments, or indeed, any arguments at all. Look at the following table.

Text Mode Switches
Use `\normalfont` to switch to the default normal font.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\rmfamily` to switch to roman letters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\sffamily` to switch to sans serif latters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\ttfamily` to switch to typewriter letters.
`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`
Use `\mdseries` to switch to medium letters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\bfseries` to switch to boldface letters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\upshape` to switch to upright letters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\itshape` to switch to italic letters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\slshape` to switch to slanted letters.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
Use `\scshape` to switch to small capital letters.
ABCDEFGHIJKLMNOPQRSTUVWXYZABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789

To illustrate how these switches work, the following unindented paragraph is in the default text font (it's just filler text).
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis egestas urna et dolor posuere, accumsan faucibus justo viverra. Pellentesque libero neque, maximus vitae placerat eu, luctus sit amet sapien. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aenean vel nisl massa. Sed id velit tellus. Vivamus eu elit a erat aliquam auctor nec sed mi. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Cras ut consequat purus. Fusce imperdiet scelerisque sagittis. Etiam maximus sagittis sapien. Proin mi metus, cursus a ex eget, maximus laoreet sapien. In hac habitasse platea dictumst. Morbi eget est dui. Cras posuere nisl quis leo facilisis, vel viverra augue sagittis. Nullam posuere, ex id efficitur mollis, velit nisi tincidunt augue, vel tempor orci nisi a justo.

The following unindented paragraph is the same filler text, but now preceded by the `\sffamily` switch.
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis egestas urna et dolor posuere, accumsan faucibus justo viverra. Pellentesque libero neque, maximus vitae placerat eu, luctus sit amet sapien. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aenean vel nisl massa. Sed id velit tellus. Vivamus eu elit a erat aliquam auctor nec sed mi. Orci varius natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Cras ut consequat purus. Fusce imperdiet scelerisque sagittis. Etiam maximus sagittis sapien. Proin mi metus, cursus a ex eget, maximus laoreet sapien. In hac habitasse platea dictumst. Morbi eget est dui. Cras posuere nisl quis leo facilisis, vel viverra augue sagittis. Nullam posuere, ex id efficitur mollis, velit nisi tincidunt augue, vel tempor orci nisi a justo.

Note that this (the current) paragraph is still typeset in sans serif letters because the `\sffamily` switch is still activated. We can deactivate it by activating another switch, like `\normalfont` to get back to normal default text, or like `\bfseries` **to get bold letters, or even like** `\itshape` ***to get italic letters. Note that we got both bold and italic letters in that last phrase, and in this sentence, because the*** `\bfseries` ***switch was still activated when we activated the*** `\itshape` ***switch. Activated switches can accumulate to give us new combinations of letter styles and shapes. We can return to the normal default font by activating the*** `\normalfont` ***switch*** just like this. Now everything is back to normal.

1.2 Math Mode Fonts

Now we need to discuss math mode fonts, which are considerably more complicated than text mode fonts. There are math mode font *commands*, but there are no math mode font *switches*. Math mode fonts sometimes include lowercase and/or uppercase Greek letters, which are frequently used in mathematical notation. Math mode fonts go into effect whenever you enter in-line math mode with `\(...\)` or display math mode with `\[...\]`. Many LaTeX tutorials use `$...$` for in-line math mode and `$$...$$` for display math mode. For various technical reasons, these are not recommended so please do not use them. Please consistently use `\(...\)` for in-line math mode and `\[...\]` for display math mode.

1.2.1 Unicode Math Mode Font Commands

In math mode, we use the `unicode-math` package to create new font combinations that allow for greater flexibility than otherwise available. There is a tradeoff, and that is documents must be processed with the LuaLaTeX engine. The math mode `unicode` font commands available to you in this habitat are given in the following table. Note that all of these commands begin with `\sym<...>`. You may assume that characters not shown are not supported.

`\(...\)` or `\[...\]` or `\symnormal{...}` give the default math mode font. Use this for vector index notation.
*abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ*0123456789
αβγδϵεζηθϑικλμνξοπϖρϱσςτυϕφχψωΔΓΘΛΞΠΣΥΦΨΩ
`\symbf{...}` gives boldface italic. Use this for coordinate-free vectors and matrices.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
αβγδϵεζηθϑικλμνξοπϖρϱσςτυϕφχψωΔΓΘΛΞΠΣΥΦΨΩ
`\symup{...}` gives upright (roman) serif. Use this to name particles. `\symup{\Omega}` (Ω) represents the ohm.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
αβγδϵεζηθϑικλμνξοπϖρϱσςτυϕφχψωΔΓΘΛΞΠΣΥΦΨΩ
`\symbfup{...}` gives boldface upright serif.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
αβγδϵεζηθϑικλμνξοπϖρϱσςτυϕφχψωΔΓΘΛΞΠΣΥΦΨΩ
`\symsfup{...}` gives sans serif upright. Use this for physical dimensions.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
`\symbfsfup{...}` gives boldface sans serif upright.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789
αβγδϵεζηθϑικλμνξοπϖρϱσςτυϕφχψωΔΓΘΛΞΠΣΥΦΨΩ
`\symsfit{...}` gives sans serif italic. Use this for tensor index notation.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
`\symbfsfit{...}` gives boldface sans serif italic. Use this for coordinate-free tensors.
abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
`\symcal{...}` and`\symbfcal{...}` give calligraphic. Use these for points on a manifold or naming coordinate systems.
𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵
𝓐𝓑𝓒𝓓𝓔𝓕𝓖𝓗𝓘𝓙𝓚𝓛𝓜𝓝𝓞𝓟𝓠𝓡𝓢𝓣𝓤𝓥𝓦𝓧𝓨𝓩
`\symscr{...}` and `\symbfscr{...}` give script. Use these for naming spacetime events.
𝒶𝒷𝒸𝒹ℯ𝒻ℊ𝒽𝒾𝒿𝓀𝓁𝓂𝓃ℴ𝓅𝓆𝓇𝓈𝓉𝓊𝓋𝓌𝓍𝓎𝓏𝒜ℬ𝒞𝒟ℰℱ𝒢ℋℐ𝒥𝒦ℒℳ𝒩𝒪𝒫𝒬ℛ𝒮𝒯𝒰𝒱𝒲𝒳𝒴𝒵
𝓪𝓫𝓬𝓭𝓮𝓯𝓰𝓱𝓲𝓳𝓴𝓵𝓶𝓷𝓸𝓹𝓺𝓻𝓼𝓽𝓾𝓿𝔀𝔁𝔂𝔃𝓐𝓑𝓒𝓓𝓔𝓕𝓖𝓗𝓘𝓙𝓚𝓛𝓜𝓝𝓞𝓟𝓠𝓡𝓢𝓣𝓤𝓥𝓦𝓧𝓨𝓩
`\symtt{...}` gives typewriter.
`abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789`
`\symfrak{...}` and `\symbffrak{...}` give Fraktur.
𝔞𝔟𝔠𝔡𝔢𝔣𝔤𝔥𝔦𝔧𝔨𝔩𝔪𝔫𝔬𝔭𝔮𝔯𝔰𝔱𝔲𝔳𝔴𝔵𝔶𝔷𝔄𝔅ℭ𝔇𝔈𝔉𝔊ℌℑ𝔍𝔎𝔏𝔐𝔑𝔒𝔓𝔔ℜ𝔖𝔗𝔘𝔙𝔚𝔛𝔜ℨ
𝖆𝖇𝖈𝖉𝖊𝖋𝖌𝖍𝖎𝖏𝖐𝖑𝖒𝖓𝖔𝖕𝖖𝖗𝖘𝖙𝖚𝖛𝖜𝖝𝖞𝖟𝕬𝕭𝕮𝕯𝕰𝕱𝕲𝕳𝕴𝕵𝕶𝕷𝕸𝕹𝕺𝕻𝕼𝕽𝕾𝕿𝖀𝖁𝖂𝖃𝖄𝖅
`\symbb{...}` gives blackboard.
𝕒𝕓𝕔𝕕𝕖𝕗𝕘𝕙𝕚𝕛𝕜𝕝𝕞𝕟𝕠𝕡𝕢𝕣𝕤𝕥𝕦𝕧𝕨𝕩𝕪𝕫𝔸𝔹ℂ𝔻𝔼𝔽𝔾ℍ𝕀𝕁𝕂𝕃𝕄ℕ𝕆ℙℚℝ𝕊𝕋𝕌𝕍𝕎𝕏𝕐ℤ𝟘𝟙𝟚𝟛𝟜𝟝𝟞𝟟𝟠𝟡
`\symbbit{...}` gives blackboard italic. I have no idea why it's defined for only five letters.
ⅆⅇⅈⅉⅅ

The zero vector is `\symbfup{0}` or **0**. It is aliased as `\zerovector`, which also gives **0**. As you can see, many more characters are available using the `unicode` commands. Also, notice the semantic naming of each typeface and style.

1.2.2 A Few More Mathematical Symbols

There's a handful of other mathematical symbols you will need, and here they are. Don't worry if you've never seen any or all of these before. The third row of symbols is unique to `mandi`.

$\nabla \, \square \, \partial \; \otimes \; \times \; \cdot \; \bullet$
$\boldsymbol{\nabla} \; \boldsymbol{\partial}$
$\nabla_{\boldsymbol{u}} \, \nabla_{\boldsymbol{u}}$
$\underline{\qquad} \; \underline{\;\boldsymbol{a}\;} \, \mathbb{C}_{(1,2)} \, \mathbb{C}_{(1,3)(2,4)} \, \mathsf{C}_{(1,2)} \, \mathsf{C}_{(1,3)(2,4)}$

1.2.3 Unicode Input

With Unicode support, you can use characters direction from your keyboard. The following example of Maxwell's equations uses keyboard characters typed directly.

$$\nabla \bullet \boldsymbol{E} = \frac{\rho}{\varepsilon_o}$$
$$\nabla \bullet \boldsymbol{B} = 0$$
$$\nabla \times \boldsymbol{E} = -\frac{\partial \boldsymbol{B}}{\partial t}$$
$$\nabla \times \boldsymbol{B} = \mu_o\left(I + \varepsilon_o \frac{\partial \boldsymbol{E}}{\partial t}\right)$$

There is an intelligent exterior derivative operator.

$$\mathbf{d}\omega$$

2 The Unit Engine

The seven fundamental SI quantities are displacement ($3\ \mathtt{m}$), time ($3\ \mathtt{s}$), electric current ($1.2\ \mathtt{A}$), thermodynamic temperature ($3\ \mathtt{K}$), amount ($4\ \mathtt{mol}$), and luminous intensity ($2\ \mathtt{cd}$). Momentum ($4\ \mathtt{N}\cdot\mathtt{s}$) is a very important quantity.

$4\ \mathtt{kg}\cdot\mathtt{m/s}$

3 Column and Row Vectors

3.1 Column Vectors

$$\begin{pmatrix}0\\1\\2\\3\end{pmatrix}\begin{pmatrix}a_1\\a_2\\a_2\end{pmatrix}$$

3.2 Row Vectors

$$\begin{pmatrix}0 & 1 & 2 & 3\end{pmatrix}\begin{pmatrix}a_1 & a_2 & a_2\end{pmatrix}$$

4 Symbolic Vectors

4.1 Notation Conventions

Typesetting symbolic vectors is ubiquitous in in physics, yet most sources are not consistent in notation compared to other sources. Using arrows to denote vector quantities, as with `\vec*{p}` (note the `*`), which gives $\vec{p}$, seems the standard in introductory physics courses, but this isn't seen in higher level resources, and certainly not in published articles. The standard, which doesn't seem to be observed, is that vectors are typeset in boldface italic, as with `\vec{p}`, which gives $\boldsymbol{p}$. Physics teachers know students struggle with using arrows to denote vector quantities, and most students never correctly learn it. After all, why should they learn something that isn't used consistently, or even at all? There is really just one practical reason for using arrows rather than boldface, and that is the quaint notion of writing on a whiteboard (or a chalkboard if anyone remembers those). We really should move with the times and enforce the professional convention for our students. So in `mandi` we do exactly this, but allow flexibility in notation for those who insist upon having it. By default, the standard LaTeX `\vec{...}` command is redefined to, by default, typeset symbols for vector quantities in boldface italic. The starred form of the command, `\vec*{...}`, typesets them with arrows. Note that getting arrow notation is a bit more difficult (because you must type the `*`) than the default boldface italic notation, and this is a subtle psychological tactic on my part; the new notation takes less effort.

4.2 Symbolic Notation

The command `\dirvect{...}` typesets the symbol for a vector's direction. `\dirvect{r}` gives $\widehat{\boldsymbol{r}}$ and as you might expect by now `\dirvect*{r}` gives $\widehat{r}$. A vector's magnitude is typeset with `\norm{...}`.

As you may predict, `\norm{\vec{r}}` gives $\|\boldsymbol{r}\|$ and `\norm{\vec*{r}}` gives $\|\vec{r}\|$. Any vector can be *factored* into its magnitude and direction. So $\boldsymbol{p} = \|\boldsymbol{p}\|\widehat{\boldsymbol{p}}$ or $\vec{p} = \|\vec{p}\|\widehat{p}$.

The command `\magvect{...}` typesets the symbol for a vector's magnitude. `\magvect{r}` gives $\|\boldsymbol{r}\|$ and of course `\magvect*{r}` gives $\|\vec{r}\|$.

Finally, the command `\Dvect{...}` typesets the symbol for the change in a vector. `\Dvect{r}` gives $\Delta\boldsymbol{r}$. For arrow notation, use the starred form. `\Dvect*{r}` gives $\Delta\vec{r}$.

4.3 Embellishments

Again, I state that notation should help the reader and for physics students, unambiguously help guide one's thinking. This frequently requires use of embellishments, or additions to basic notation. For example, consider a rock moving through space. One can speak of the rock's momentum, and notate it as `\vec{p}_{rock}`, which gives

$$\boldsymbol{p}_{rock}$$

as predicted. But wait, the subscript doesn't look right. It is typeset in italic when it shouldn't be. So instead, we write `\vec{p}_{\mathup{rock}}`, which gives

$$\boldsymbol{p}_{\mathrm{rock}}$$

which looks much better. Why does this make a difference? The answer is that the `\vec{...}` command internally takes place in *math mode*, which means all characters are, by default, typeset in italic. Words are not intended to be written in math mode. So if we want to typeset an actual piece of text, you must mark it up as such using a `\mathup{...}` command.

So now you can correctly typeset an initial momentum and final momentum symbolically as `\vec{p}_{\mathup{initial}}` and `\vec{p}_{\mathup{final}}`, which give

$$\boldsymbol{p}_{\mathrm{initial}} \text{ and } \boldsymbol{p}_{\mathrm{final}}$$

exactly as expected. If you want something other than actual text as a subscript, just supply it without additional markup. So you could write, for example, `\vec{p}_{1}` and get

$$\boldsymbol{p}_{1}$$

as expected. The braces around the subscript are strictly not required if the subscript is just one character, but it is a good habit to write them all the time, so I do.

You can add superscripts just as easily as you can add subscripts, using a caret (^) to denote the superscript. Writing `\vec{p}^{\mathup{rock}}` gives

$$\boldsymbol{p}^{\mathrm{rock}}$$

just like that. All the comments about subscripts above apply to superscripts. The best part is that you can effortlessly mix subscripts and superscripts. Writing
`\vec{p}_{\mathup{final}}^{\mathup{rock}}` or
`\vec{p}^{\mathup{rock}}_{\mathup{final}}` gives

$$\boldsymbol{p}_{\mathrm{final}}^{\mathrm{rock}} \text{ and } \boldsymbol{p}^{\mathrm{rock}}_{\mathrm{final}}$$

and note that the order of the flourishes doesn't matter. The underlying LaTeX3 programming layer makes this almost trivial to implement in `mandi` now. Using text to embellish a symbolic vector takes a bit more effort, which is a subtle warning to use it sparingly. Simply changing `\vec{...}` to `\vec*{...}` gives the same results with arrow notation and gives

$$\vec{p}_{\mathrm{final}}^{\mathrm{rock}} \text{ and } \vec{p}_{\mathrm{final}}^{\mathrm{rock}}$$

as expected.

It is very important to always that when in math mode, always, without exception, wrap words used in subscripts and superscripts in `\mathup{...}`. This does not apply to mathematical symbols, only to actual words or any text that is not mathematical notation.

Of course you can mix mathematical notation and text in the flourishes. As an example, one could represent an alpha particle's final momentum by writing
`\vec{p}_{\symup{\alpha},\mathup{final}}` which typesets as

$$\boldsymbol{p}_{\mathrm{\alpha},\mathrm{final}}$$

as beautifully as expected. If you want arrow notation, you would write
`\vec*{p}_{\symup{\alpha},\mathup{final}}` which typesets as

$$\vec{p}_{\mathrm{\alpha},\mathrm{final}}$$

as expected. For many reasons, I think the boldface notation looks better. Finally, note that `\symup{...}` is used here since upright Greek letters are used to name subatomic particles.

$$\left\lVert \boldsymbol{p}_{\mathrm{final}} \right\rVert \qquad \left(\frac{1}{4\pi\varepsilon_o}\right)$$

$$\mathit{\Delta}\boldsymbol{r} = \boldsymbol{r}_{\mathrm{final}} - \boldsymbol{r}_{\mathrm{initial}} \qquad \text{definition} \tag{1}$$

$$\mathit{\Delta}\vec{r} = \vec{r}_{\mathrm{final}} - \vec{r}_{\mathrm{initial}} \qquad \text{Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi.} \tag{2}$$

$$\mathit{\Delta}\boldsymbol{r} = \boldsymbol{r}_{\mathrm{final}} - \boldsymbol{r}_{\mathrm{initial}} \qquad \text{definition} \tag{3}$$

$$\mathit{\Delta}\boldsymbol{r} = \boldsymbol{r}_{\mathrm{final}} - \boldsymbol{r}_{\mathrm{initial}} \qquad \text{definition} \tag{4}$$

$$\lVert\boldsymbol{r}\rVert = \sqrt{\boldsymbol{r}\cdot\boldsymbol{r}} \tag{5}$$

$$\left(\frac{1}{4\pi\epsilon_o}\right) \qquad (\,\cdot\,)$$
$$\left[\frac{1}{4\pi\epsilon_o}\right] \qquad [\,\cdot\,]$$
$$\left|\frac{1}{4\pi\epsilon_o}\right| \qquad |\,\cdot\,|$$
$$\left\|\frac{1}{4\pi\epsilon_o}\right\| \qquad \|\,\cdot\,\|$$
$$\left(\frac{1}{4\pi\epsilon_o}\right) \qquad (\,\cdot\,)$$
$$\left[\frac{1}{4\pi\epsilon_o}\right] \qquad [\,\cdot\,]$$
$$\left|\frac{1}{4\pi\epsilon_o}\right| \qquad |\,\cdot\,|$$
$$\left\|\frac{1}{4\pi\epsilon_o}\right\| \qquad \|\,\cdot\,\|$$

4.4 Coordinate-Free Notation

The arrow notation used in introductory physics is a compromise between a notation that ostensibly helps students remember the quantity in question had an associate direction and a notation that is easily reproducible by hand on a whiteboard. As hard as we try, students rarely use arrow notation for vectors correctly, if at all. You don't find arrow notation for vectors much at all in the literature outside of introductory textbooks. Most sources use boldface, and while that is harder to use on a whiteboard, it's trivial to do in LaTeX so why not adopt it as the expectation?

Use the `\veccomp{...}` command to get coordinate-free notation for vectors. `\veccomp{p}` gives $\boldsymbol{p}$. This notation isn't intended to be used with embellishments, but it can be done as long as you are in math mode.

4.5 Index Notation

Use the `\veccomp*{...}` command to get index notation for vectors. `\veccomp*{p}` gives p. Where is the index? You have to supply it using the `\indices{...}` command from the `tensor` package, which `mandi` loads for you. So to get the symbol for momentum in index notation you would use `\veccomp*{p}\indices{^i}` which gives p^i. Now, you may realize that `\veccomp*{p}^i` gives the same result, and it does indeed give p^i as expected, so why bother with the `\indices{...}` command? The answer is that the `\indices{...}` command plays an important role in tensor index notation, which includes vector index notation, and you need to be comfortable using it. Use `\veccomp{e}` to get a basis vector, which gives $\boldsymbol{e}$ as expected. Basis vectors must be labeled, and I recommend using the `indices{...}` command, but both `\veccomp{e}\indices{_i}` and `\veccomp{e}_i` give the same result, $\boldsymbol{e}_i$ as expected. So to get the full symbol for a vector in a particular basis you would use

`\veccomp*{p}_i\veccomp{e}^i` or `\veccomp*{p}\indices{^i}\veccomp{e}\indices{_i}`, both of which give $p^i \boldsymbol{e}_i$.

5 Highlighting Mathematics

A new feature to `mandi` is a command for highlighting mathematics.

$$
\begin{aligned}
(\Delta s)^2 &= -(\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2 \\
(\Delta s)^2 &= -(\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2 \\
(\Delta s)^2 &= -(\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2 \\
(\Delta s)^2 &= -(\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2 \\
(\Delta s)^2 &= -(\Delta t)^2 + (\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2
\end{aligned}
$$

Here is another example.

$$
\begin{aligned}
\Delta\mathbf{p} &= \mathbf{F}_{\text{net,sys}}\,\Delta t \\
\Delta\mathbf{p} &= \mathbf{F}_{\text{net,sys}}\,\Delta t \\
\Delta\mathbf{p} &= \mathbf{F}_{\text{net,sys}}\,\Delta t \\
\Delta\mathbf{p} &= \mathbf{F}_{\text{net,sys}}\,\Delta t \\
\Delta\mathbf{p} &= \mathbf{F}_{\text{net,sys}}\,\Delta t \\
\Delta\mathbf{p} &= \mathbf{F}_{\text{net,sys}}\,\Delta t
\end{aligned}
$$

Highlighting also works in in-line math mode like $a^2 + b^2 = c^2$.

6 Tensor Notation

A tensor has a valence $\binom{m}{n}$ by definition. A vector is a $\binom{1}{0}$ tensor and a one-form is a $\binom{0}{1}$ tensor.

7 `GlowScript` and `VPython` Program Listings

7.1 The `glowscriptblock` Environment

`GlowScript` program listings are improved and look better. Clicking anywhere on the actual code takes the reader to the relevant source code on GlowScript.org for read-only access using the URL provided by the user. To minimize the chances of a program being broken over a pagebreak, each listing begins on a new page. This is a big change from the previous version of `mandi` and may be annoying for small programs, but I think the clarity of beginning a new listing on a new page is important.

```
GlowScript 3.0 VPython

scene.width = 400
scene.height = 760
# constants and data
g = 9.8        # m/s^2
mball = 0.03   # kg
Lo = 0.26      # m
ks = 1.8       # N/m
deltat = 0.01 #s

# objects (origin is at ceiling)
ceiling = box(pos=vector(0,0,0), length=0.2, height=0.01, width=0.2)
ball = sphere(pos=vector(0,-0.3,0),radius=0.025,color=color.orange)
spring = helix(pos=ceiling.pos, axis=ball.pos-ceiling.pos,
         color=color.cyan,thickness=0.003,coils=40,radius=0.010)

# initial values
pball = mball * vector(0,0,0)      # kg m/s
Fgrav = mball * g * vector(0,-1,0) # N
t = 0

# improve the display
scene.autoscale = False        # turn off automatic camera zoom
scene.center = vector(0,-Lo,0) # move camera down
scene.waitfor('click')         # wait for a mouse click

# initial calculation loop
# calculation loop
#while t < 10:
#    rate(100)
#    Fnet = Fgrav
#    pball = pball + Fnet * deltat
#    ball.pos = ball.pos + (pball / mball) * deltat
#    spring.axis = ball.pos - ceiling.pos
#    t = t + deltat

# modified initial calculation loop
# calculation loop
while t < 10:
    rate(100)
    # we need the stretch
    s = mag(ball.pos) - Lo
    # we need the spring force
    Fspring = ks * s * -norm(spring.axis)
    Fnet = Fgrav + Fspring
    pball = pball + Fnet * deltat
    ball.pos = ball.pos + (pball / mball) * deltat
    spring.axis = ball.pos - ceiling.pos
    t = t + deltat
```

Listing 1: A `GlowScript` program.

nameref-link to listing: A `GlowScript` program
ref-link to listing: 1
pageref-link to listing: 13
autoref-link to listing: Listing 1

It may be annoying for a small program listing like this, but again, I think the clarity is important, especially for beginners.

```
1234567890123456789012345678901234567890123456789012345678901234567890123456789
1234567890123456789012345678901234567890123456789012345678901234567890123456789
```

Listing 2: 79 Columns

nameref-link to listing: 79 Columns
ref-link to listing: 2
pageref-link to listing: 15
autoref-link to listing: Listing 2

7.2 The `glowscriptblock` Environment With `VPython` Programs

`VPython` programs are also nicely formatted but lack the hyperlink capability because they live on the user's local computer and not online. Since the `glowscriptblock` environment turns the code into a hyperlink, it makes no sense semantically to use this environment for `VPython` programs. Still, it is sometimes helpful to show a block of code this way without turning the code into a hyperlink. Use `glowscriptblock` without the optional argument to typeset a block of `VPython` code. `\begin{glowscriptblock}{A \VPython\ program}{pref2}` gives

```python
from vpython import *

scene.width = 400
scene.height = 760
# constants and data
g = 9.8       # m/s^2
mball = 0.03  # kg
Lo = 0.26     # m
ks = 1.8      # N/m
deltat = 0.01 #s

# objects (origin is at ceiling)
ceiling = box(pos=vector(0,0,0), length=0.2, height=0.01, width=0.2)
ball = sphere(pos=vector(0,-0.3,0),radius=0.025,color=color.orange)
spring = helix(pos=ceiling.pos, axis=ball.pos-ceiling.pos,
         color=color.cyan,thickness=0.003,coils=40,radius=0.010)

# initial values
pball = mball * vector(0,0,0)      # kg m/s
Fgrav = mball * g * vector(0,-1,0) # N
t = 0

# improve the display
scene.autoscale = False        # turn off automatic camera zoom
scene.center = vector(0,-Lo,0) # move camera down
scene.waitfor('click')         # wait for a mouse click

# initial calculation loop
# calculation loop
#while t < 10:
#    rate(100)
#    Fnet = Fgrav
#    pball = pball + Fnet * deltat
#    ball.pos = ball.pos + (pball / mball) * deltat
#    spring.axis = ball.pos - ceiling.pos
#    t = t + deltat

# modified initial calculation loop
# calculation loop
while t < 10:
    rate(100)
    # we need the stretch
    s = mag(ball.pos) - Lo
    # we need the spring force
    Fspring = ks * s * -norm(spring.axis)
    Fnet = Fgrav + Fspring
    pball = pball + Fnet * deltat
    ball.pos = ball.pos + (pball / mball) * deltat
    spring.axis = ball.pos - ceiling.pos
    t = t + deltat
```

Listing 3: A `VPython` program

nameref-link to listing: A `VPython` program
ref-link to listing: 3
pageref-link to listing: 16
autoref-link to listing: Listing 3

7.3 In-line `GlowScript` and `VPython` Code

In-line code can be included as before using `\glowscriptline{...}` or `\vpythonline{...}`, depending on which is semantically appropriate. Using
`\glowscriptline{Earth = sphere(color=color.blue)}` gives
`Earth = sphere(pos=vector(1,1,1),radius=2,color=color.blue)` and
`\vpythonline{Sun = sphere(pos=vector(1,1,1),radius=2,color=color.yellow)}` gives
`Sun = sphere(pos=vector(1,1,1),radius=2,color=color.yellow)`.

7.4 Including a `VPython` File

A `VPython` file can be included using `\vpythonfile{...}{...}{...}`. The file should be in the same folder as the document being typeset, and a caption and a reference label must be provided. Again, the typeset listing begins on a new page.

```python
from vpython import *

scene.width = 400
scene.height = 760
# constants and data
g = 9.8       # m/s^2
mball = 0.03  # kg
Lo = 0.26     # m
ks = 1.8      # N/m
deltat = 0.01 #s

# objects (origin is at ceiling)
ceiling = box(pos=vector(0,0,0), length=0.2, height=0.01, width=0.2)
ball = sphere(pos=vector(0,-0.3,0),radius=0.025,color=color.orange)
spring = helix(pos=ceiling.pos, axis=ball.pos-ceiling.pos,
         color=color.cyan,thickness=0.003,coils=40,radius=0.010)

# initial values
pball = mball * vector(0,0,0)      # kg m/s
Fgrav = mball * g * vector(0,-1,0) # N
t = 0

# improve the display
scene.autoscale = False        # turn off automatic camera zoom
scene.center = vector(0,-Lo,0) # move camera down
scene.waitfor('click')         # wait for a mouse click

# initial calculation loop
# calculation loop
#while t < 10:
#    rate(100)
#    Fnet = Fgrav
#    pball = pball + Fnet * deltat
#    ball.pos = ball.pos + (pball / mball) * deltat
#    spring.axis = ball.pos - ceiling.pos
#    t = t + deltat

# modified initial calculation loop
# calculation loop
while t < 10:
    rate(100)
    # we need the stretch
    s = mag(ball.pos) - Lo
    # we need the spring force
    Fspring = ks * s * -norm(spring.axis)
    Fnet = Fgrav + Fspring
    pball = pball + Fnet * deltat
    ball.pos = ball.pos + (pball / mball) * deltat
    spring.axis = ball.pos - ceiling.pos
    t = t + deltat
```

Listing 4: Sample Program

nameref-link to listing: Sample Program
ref-link to listing: 4
pageref-link to listing: 18
autoref-link to listing: Listing 4

8 Commands Specific to *Matter & Interactions*

These commands are defined in an optional accessory package called `mandiexp`. That package's name is subject to change.

$\Delta\boldsymbol{p}_{\text{sys}} = \boldsymbol{F}_{\text{sys,net}}\,\Delta t$	momentum principle
$\boldsymbol{p}_{\text{sys,final}} = \boldsymbol{p}_{\text{sys,initial}} + \boldsymbol{F}_{\text{sys,net}}\,\Delta t$	momentum principle, update form
$\Delta E_{\text{sys}} = W_{\text{ext}} + Q$	energy principle
$E_{\text{sys,final}} = E_{\text{sys,initial}} + W_{\text{ext}} + Q$	energy principle, update form
$\Delta\boldsymbol{L}_{A,\text{sys}} = \boldsymbol{\tau}_{A,\text{sys,net}}\,\Delta t$	angular momentum principle
$\boldsymbol{L}_{A,\text{sys,final}} = \boldsymbol{L}_{A,\text{sys,initial}} + \boldsymbol{\tau}_{A,\text{sys,net}}\,\Delta t$	angular momentum principle, update form

E_{sys}	$E_{\text{sys,final}}$	$E_{\text{sys,initial}}$	ΔE_{sys}
E_{particle}	$E_{\text{particle,final}}$	$E_{\text{particle,initial}}$	$\Delta E_{\text{particle}}$
E_{rest}	$E_{\text{rest,final}}$	$E_{\text{rest,initial}}$	ΔE_{rest}
E_{internal}	$E_{\text{internal,final}}$	$E_{\text{internal,initial}}$	$\Delta E_{\text{internal}}$
E_{chem}	$E_{\text{chem,final}}$	$E_{\text{chem,initial}}$	ΔE_{chem}
E_{therm}	$E_{\text{therm,final}}$	$E_{\text{therm,initial}}$	ΔE_{therm}
E_{photon}	$E_{\text{photon,final}}$	$E_{\text{photon,initial}}$	ΔE_{photon}
E_{electron}	$E_{\text{electron,final}}$	$E_{\text{electron,initial}}$	$\Delta E_{\text{electron}}$
$E_{\upbeta}$	$E_{\upbeta\text{,final}}$	$E_{\upbeta\text{,initial}}$	$\Delta E_{\upbeta}$
K_{trans}	$K_{\text{trans,final}}$	$K_{\text{trans,initial}}$	ΔK_{trans}
E_{K}	$E_{\text{K,final}}$	$E_{\text{K,initial}}$	ΔE_{K}
K_{rot}	$K_{\text{rot,final}}$	$K_{\text{rot,initial}}$	ΔK_{rot}
E_{rot}	$E_{\text{rot,final}}$	$E_{\text{rot,initial}}$	ΔE_{rot}
K_{vib}	$K_{\text{vib,final}}$	$K_{\text{vib,initial}}$	ΔK_{vib}
E_{vib}	$E_{\text{vib,final}}$	$E_{\text{vib,initial}}$	ΔE_{vib}
U_{g}	$U_{\text{g,final}}$	$U_{\text{g,initial}}$	ΔU_{g}
U_{e}	$U_{\text{e,final}}$	$U_{\text{e,initial}}$	ΔU_{e}
U_{s}	$U_{\text{s,final}}$	$U_{\text{s,initial}}$	ΔU_{s}

These commands are intended to be semantically helpful.