Your Thesis

The title of your very nice and highly important, never before seen scientific work of stellar excellence

Your Name

1st Examiner	*Prof. Alfreb Einstime*
2nd Examiner	*Prof. Rufols Diebels*
Supervisor	*Unimportant Person, M.Sc.*

29th May 2019

`yatt_thesis n.a. (n.a.)`

LuaTeX, Version 1.10.0 (TeX Live 2019/W32TeX).
LaTeX 2_ε (2018-12-01) with `glossaries-extra` (`bib2gls`) and biber, using `latexmk` with a `splitidx` subroutine by J. Collins.

Dear Your Name,
registration number 666.

Your Thesis Task.

Topic:
The title of your very nice and highly important, never before seen scientific work of stellar excellence

As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental unity of apperception can not take account of the discipline of natural reason, by means of analytic unity. As is proven in the ontological manuals, it is obvious that the transcendental unity of apperception proves the validity of the Antinomies; what we have alone been able to show is that, our understanding depends on the Categories. It remains a mystery why the Ideal stands in need of reason. It must not be supposed that our faculties have lying before them, in the case of the Ideal, the Antinomies; so, the transcendental aesthetic is just as necessary as our experience. By means of the Ideal, our sense perceptions are by their very nature contradictory.

- First you are doing this,
- then another thing,
- lastly that.

You will be working on this a lot.

Fantasy land, October 2020

Rest of this page intentionally left blank.

Declaration of Authorship

I, Your Name [666], hereby declare to be the sole, independent author of the submitted here. No other than the cited references have been used. Any content directly or indirectly obtained from external sources has been marked up as such. This thesis has neither been submitted to a second examination authority nor been published.

Your Name	*Place & Date*

Rest of this page intentionally left blank.

Abstract

This is the public version with some redacted information
As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Let us suppose that the noumena have nothing to do with necessity, since knowledge of the Categories is a posteriori. Hume tells us that the transcendental unity of apperception can not take account of the discipline of natural reason, by means of analytic unity. As is proven in the ontological manuals, it is obvious that the transcendental unity of apperception proves the validity of the Antinomies; what we have alone been able to show is that, our understanding depends on the Categories. It remains a mystery why the Ideal stands in need of reason. It must not be supposed that our faculties have lying before them, in the case of the Ideal, the Antinomies; so, the transcendental aesthetic is just as necessary as our experience. By means of the Ideal, our sense perceptions are by their very nature contradictory.

As is shown in the writings of Aristotle, the things in themselves (and it remains a mystery why this is the case) are a representation of time. Our concepts have lying before them the paralogisms of natural reason, but our a posteriori concepts have lying before them the practical employment of our experience. Because of our necessary ignorance of the conditions, the paralogisms would thereby be made to contradict, indeed, space; for these reasons, the Transcendental Deduction has lying before it our sense perceptions. (Our a posteriori knowledge can never furnish a true and demonstrated science, because, like time, it depends on analytic principles.) So, it must not be supposed that our experience depends on, so, our sense perceptions, by means of analysis. Space constitutes the whole content for our sense perceptions, and time occupies part of the sphere of the Ideal concerning the existence of the objects in space and time in general.

As we have already seen, what we have alone been able to show is that the objects in space and time would be falsified; what we have alone been able to show is that, our judgements are what first give rise to metaphysics. As I have shown elsewhere, Aristotle tells us that the objects in space and time, in the full sense of these terms, would be

falsified. Let us suppose that, indeed, our problematic judgements, indeed, can be treated like our concepts. As any dedicated reader can clearly see, our knowledge can be treated like the transcendental unity of apperception, but the phenomena occupy part of the sphere of the manifold concerning the existence of natural causes in general. Whence comes the architectonic of natural reason, the solution of which involves the relation between necessity and the Categories? Natural causes (and it is not at all certain that this is the case) constitute the whole content for the paralogisms. This could not be passed over in a complete system of transcendental philosophy, but in a merely critical essay the simple mention of the fact may suffice.

Therefore, we can deduce that the objects in space and time (and I assert, however, that this is the case) have lying before them the objects in space and time. Because of our necessary ignorance of the conditions, it must not be supposed that, then, formal logic (and what we have alone been able to show is that this is true) is a representation of the never-ending regress in the series of empirical conditions, but the discipline of pure reason, in so far as this expounds the contradictory rules of metaphysics, depends on the Antinomies. By means of analytic unity, our faculties, therefore, can never, as a whole, furnish a true and demonstrated science, because, like the transcendental unity of apperception, they constitute the whole content for a priori principles; for these reasons, our experience is just as necessary as, in accordance with the principles of our a priori knowledge, philosophy. The objects in space and time abstract from all content of knowledge. Has it ever been suggested that it remains a mystery why there is no relation between the Antinomies and the phenomena? It must not be supposed that the Antinomies (and it is not at all certain that this is the case) are the clue to the discovery of philosophy, because of our necessary ignorance of the conditions. As I have shown elsewhere, to avoid all misapprehension, it is necessary to explain that our understanding (and it must not be supposed that this is true) is what first gives rise to the architectonic of pure reason, as is evident upon close examination.

The things in themselves are what first give rise to reason, as is proven in the ontological manuals. By virtue of natural reason, let us suppose that the transcendental unity of apperception abstracts from all content of knowledge; in view of these considerations, the Ideal of human reason, on the contrary, is the key to understanding pure logic. Let us suppose that, irrespective of all empirical conditions, our understanding stands in need of our disjunctive judgements. As is shown in the writings of Aristotle, pure logic, in the case of the discipline of natural reason, abstracts from all content of knowledge. Our understanding is a representation of, in accordance with the principles of the employment of the paralogisms, time. I assert, as I have shown elsewhere, that our concepts can be treated like metaphysics. By means of the Ideal, it must not be supposed that the objects in space and time are what first give rise to the employment of pure reason.

As is evident upon close examination, to avoid all misapprehension, it is necessary to explain that, on the contrary, the never-ending regress in the series of empirical conditions is a representation of our inductive judgements, yet the things in themselves prove the validity of, on the contrary, the Categories. It remains a mystery why, indeed, the never-ending regress in the series of empirical conditions exists in philosophy,

but the employment of the Antinomies, in respect of the intelligible character, can never furnish a true and demonstrated science, because, like the architectonic of pure reason, it is just as necessary as problematic principles. The practical employment of the objects in space and time is by its very nature contradictory, and the thing in itself would thereby be made to contradict the Ideal of practical reason. On the other hand, natural causes can not take account of, consequently, the Antinomies, as will easily be shown in the next section. Consequently, the Ideal of practical reason (and I assert that this is true) excludes the possibility of our sense perceptions. Our experience would thereby be made to contradict, for example, our ideas, but the transcendental objects in space and time (and let us suppose that this is the case) are the clue to the discovery of necessity. But the proof of this is a task from which we can here be absolved.

- we found out this
- and also that!

Rest of this page intentionally left blank.

Contents

Nomenclature

The most commonly used symbols are listed here, alongside their location of first occurrence. Symbols used but not listed here are clarified in their respective contexts.

Buggy: *You will notice that the majority of symbols do not have info on the location of their first use. These are all the symbols which first occurred in* ***floats****. The way this document was set up, most symbols are affected. Referencing floats is a tricky business and is apparently still faulty at the moment. In an actual document, it is extremely likely you will be introducing symbols in text or math mode, not in floats, negating this issue.*

Symbols

Greek	Description	1st Use	Unit
Ξ	wastegate position		–
α	flow angle in the absolute system		°, rad
β	flow angle in the relative system		°, rad
γ	ratio of specific heats		–
γ	mass fraction	3	–
ρ	density		$\mathrm{kg\,m^{-3}}$
τ	shear stress		$\mathrm{N\,m^{-2}}$

Roman	Description	1st Use	Unit
A	area		$\mathrm{m^2}$
$\mathbf{c}$	absolute velocity (magnitude)		$\mathrm{m\,s^{-1}}$
c_p	specific heat capacity at constant pressure		$\mathrm{J\,kg^{-1}\,K^{-1}}$
H	enthalpy		J
H	(specific) heating value	3	$\mathrm{J\,kg^{-1}}$
h	height		m
M	Mach number		–
m	mass		kg
p	pressure	2	Pa
r	radius		m

(*Symbols continued*)

R	radius ratio		–
R	specific gas constant	2	$\mathrm{J\,kg^{-1}\,K^{-1}}$
T	absolute temperature	2	K
U	internal energy		J
v	specific volume	2	$\mathrm{m^3\,kg^{-1}}$
ϑ	relative temperature		°C

Sub- and Superscripts

Subscripts

air	air
exh	exhaust
i	inferior
in	inlet
m	meridional
out	outlet
t	throat
T	turbine
θ	tangential/circumferential/blade

Acronyms

Short	Description	Pages
CI	Continuous Integration	28
HFO	Heavy Fuel Oil	23
IDE	Integrated Development Environment	4
IMO	International Maritime Organization	23
MDO	Marine Diesel Oil	23
PDF	Portable Document Format	5
RE	Rated Engine	18

Illustrative Elements

List of Figures

List of Tables

List of Illustrations

List of Code

Chapter 1

Text Features

Let us first go through some of the textual features offered by LaTeX and used in this template.

1.1 Fonts

Using high-quality fonts is one goal. This includes the fantastic TeXGyre fonts, of which the *Palatino* (by Hermann Zapf) clone *Pagella* was chosen for this template. It comes with an accompanying math font of the same name. Using both font types in conjunction is made possible through the `unicode-math` package. As such, achieving a better match of text and math fonts is nigh impossible. Both fonts are vector fonts; if LaTeX yields any warnings about font size substitutions, that is bogus.

Not only do the fonts look fantastic on their own and together, they (just as important) also feature an immensely broad support for symbols and characters, as well as font shapes and weights and combinations thereof. The latter is demonstrated in Table 1.1.

Combine all that with `microtype`, and we have absolutely gorgeous typesetting. As Jeremy Clarkson would say, 'Behoooold the magnificence':

> As any dedicated reader can clearly see, the Ideal of practical reason is a representation of, as far as I know, the things in themselves; as I have shown elsewhere, the phenomena should only be used as a canon for our understanding. The paralogisms of practical reason are what first give rise to the architectonic of practical reason. As will easily be shown in the next section, reason would thereby be made to contradict, in view of these considerations, the Ideal of practical reason, yet the manifold depends on the phenomena. Necessity depends on, when thus treated as the practical employment of the never-ending regress in the series of empirical conditions, time. Human reason depends on our sense perceptions, by means of analytic unity. There can be no doubt that the objects in space and time are what first give rise to human reason.

Notice the balanced line endings? How it was incredibly easy on the eyes, therefore easy to read? Nothing gets in the way; there is nothing to stumble over, disturbing the

Table 1.1 / Examples for font features offered by the main roman font. Notice the small vertical white space after the fourth row, nicely separating content that is slightly dissimilar.

Feature	Sample Text
Regular	The quick brown Fox jumps over the lazy Dog 13 times!
Bold	**The quick brown Fox jumps over the lazy Dog 13 times!**
Italics	*The quick brown Fox jumps over the lazy Dog 13 times!*
Bold Italics	***The quick brown Fox jumps over the lazy Dog 13 times!***
Small Capitals	The quick brown Fox jumps over the lazy Dog 13 times!
Bold SC	**The quick brown Fox jumps over the lazy Dog 13 times!**
Italics SC	*The quick brown Fox jumps over the lazy Dog 13 times!*

flow: letters and words are spaced out nicely horizontally, as are lines vertically. There is a minimal number of hyphens. That is how it is meant to be.

1.1.1 Math

What often does not occur at first glance is that many documents use text and math fonts that are very different from one another. As far as I know, Microsoft's Word has usable math typesetting and sensible default fonts for that. Yet, it cannot do what dedicated, fully-fleshed text and math fonts — different fonts, but meant for one another — can achieve. They provide a seamless transition, especially when using actual text in the math environment or vice versa, like when we go for $x \to \infty$ and then also do $\int_1^2 y^2 \, \mathrm{d}y$ or maybe $a^2 + b^2 = c^2$ (the stuff you learn at university huh), all looking rather natural. Toggle the colors in the preamble, highlighting each different font family, to see all the differences. Some more examples follow.

$$pv = RT \tag{1.1}$$

$$e^{i\pi} + 1 = 0 \tag{1.2}$$

$$\lim_{x \to \infty} \frac{\pi(x)}{x/\ln(x)} = 1 \tag{1.3}$$

$$\sum_{k=0}^{n} \binom{n}{k} = 2^n \tag{1.4}$$

$$\left[M \frac{\partial}{\partial M} + \beta(g) \frac{\partial}{\partial g} + n\gamma \right] G^n(x_1, x_2, \dots, x_n; M, g) = 0 \tag{1.5}$$

$$\Delta h_{\text{change}} = 1/2 \left(c^2_{\text{exit}} - c^2_{\text{entry}} \right) \tag{1.6}$$

Symbols Note how the symbols in Equation 1.1 are hyperlinks (leading to their definition in the glossary), courtesy of packages **hyperref** and the wonderful **glossaries-extra** working together. Having not specified anything else, they take hyper-link color. That is of course a bit unfortunate. In the final document, all links should be hidden, aka black. This is a given for print output, but I also encourage it for digital output. The visual noise introduced by colored links is quite immense. Their only point is to let users know there is something clickable — they likely figure that out themselves quickly.

Math highlighting In a very unintrusive yet also not ambivalent way, we can highlight important results, as is done in Equation 1.3. This feature is a natural part of **tcolorbox**, a very powerful package for anything color and boxes!

Chemical Reactions

Closely related to purely mathematical equations are chemical ones, as shown in Equation **R** 1.1.

$$\mathrm{C} + \mathrm{O_2} \longrightarrow \mathrm{CO_2} \qquad [\mathbf{R}\,1.1]$$

This functionality is still very basic and probably poorly executed in parts. We can also use chemical compounds invoking \chcpd, as demonstrated in Equation 1.7.

$$\frac{H_\mathrm{i}}{1 \times 10^6} = 35\gamma_\mathrm{C} + 94.3\gamma_\mathrm{H} + 10.4\gamma_\mathrm{S} + 6.3\gamma_\mathrm{N} - 10.8\gamma_\mathrm{O} - 2.44\gamma_w \qquad [1.7]$$

1.1.2 Sans-Serif

Having taken care of the main/roman font, we turn to the others. Fontin is a decent sans-serif font. Quite importantly, and in contrast to almost all other free sans-serif fonts, it comes with all the bells and whistles required for stunts, see Table 1.2. This even includes small-capitals support.

1.1.3 Mono-Spaced

Wanting to display any sort of code in the document will have you looking for a mono-spaced aka typewriter font. Inconsolata, provided by Google, will be our pick here:

```
@something
def get_metadata(table, data_start: int) -> dict:
   """Get some metadata"""
   metadata = {}  # metadata found in file header

   for i, row in enumerate(table):
      if i >= data_start:
```

```
            break
        lrow = row.lower()

    return junk
```

with Python syntax highlighting taken from this GitHub repository. Note that comments are really meant to be *italics* shape (or *slanted*), but Inconsolata does not deliver that. If you don't care, Inconsolata will be good enough. Still, Consolas is much preferred: it has *true* italics and bold italics. As far as I know, it is shipped with Windows but not free; I cannot share it here, but you will be able to get your hands on it rather easily.

So what?

If you don't care for *any* of this, don't quit just yet. The features of LaTeX will do their work silently in the background for you. You won't have to worry about any of this ever again, but can rest assured it is taken care of. **Focus on your content and impress others that way (yet still bedazzling them with your sexy-ass documents).**

1.2 References

Note that using the package **cleveref**, we only ever issue `\cref{label}` commands. The package does the heavy lifting and puts the float type in front, also with correct plural forms if required (see Table 1.1 and Figures 2.3 and 2.8 ← all of that was done automagically). Doing it any other way is just way too laborious and error-prone.

For added convenience, add some info on the type you are attaching the label to, *e.g.* `\label{fig:hello}`. This helps for auto-completion in your Integrated Development Environment (IDE)[1] of choice.

1.2.1 Bibliography

The second most likely need for references are the bibliographical ones. Examples are spread throughout this document. At their basis, they rely on **biblatex** and its back-end **biber**. Forget about **natbib** and similar. Using `\autocite{bibid}`, we can reliably cite sources and have a whole range of features delivered for free. We do not have to worry about the specific citation style (in parentheses, as a superscript, ...) — `\autocite{bibid}` takes care of that, we can then manage its behavior globally.

As such, we can have citations that look like this: (Dixon and Hall 2014, p. 23; Ronald H. Aungier 2000, pp. 29 sqq.; D. Japikse 1985, pp. 2–9; Kurzke 2011, pp. 89 sq.; Pampreen 1982). We can add a note to each; if this note is an integer number, it is

[1] Here, `gls` was used to print this abbreviation from the glossary. *Never, ever **don't*** do that: when there is an abbreviation, put it into the glossary system and have it handle that. You will regret not doing it and end up scratching together all wild, spread-out, manually typed abbreviations. Your document might suffer from inconsistencies.

Table 1.2 / Examples for font features offered by the sans-serif font.

Feature	Sample Text
Regular	The quick brown Fox jumps over the lazy Dog 13 times!
Bold	**The quick brown Fox jumps over the lazy Dog 13 times!**
Italics	*The quick brown Fox jumps over the lazy Dog 13 times!*
Bold Italics	***The quick brown Fox jumps over the lazy Dog 13 times!***
SMALL CAPITALS	THE QUICK BROWN FOX JUMPS OVER THE LAZY DOG 13 TIMES!

automatically taken to be the page number. If a following page is to be included in the citation, append `\psq`. Otherwise, `\psqq` for 'this page and the following ones'. Using `\autocites{}`, we can then chain together as many as we want.

Dixon and Hall (2014)[1] do not claim anything, this is just an example for a `\textcite`, used to implement a citation to be a readable part of the sentence. There is also a plural form available.

Language All of this is incredibly convenient, for there is minimal manual work and a lot of elevation. Importantly, this is language-agnostic, and **biblatex** will make use of **polyglossia** (the LuaLaTeX replacement for **babel**) and **csquotes**. Therefore, through changing the desired document language for just these packages, all others including **biblatex** will quickly adjust automatically.

backref Taking a look at the bibliography in the back matter of this document reveals the feature `backref`: the pages where a reference was cited occur after its entry. This is helpful in print, but amazing in digital format, for these page numbers are also links. Your readers will thank you tenfold for allowing them to very swiftly navigate and jump within your document.

Try it out by clicking this reference: (Bräunling 2015), leading you to the bibliography. It will say this page's number (Page 5) at the end of its entry, done by **biblatex**. Clicking it will land you back here exactly.

Manager To generate the `*.bib` file, from which LaTeX pulls the info, in the first place, a citation manager is a good tool. Zotero is a decent pick, but it ultimately does not matter much. Just use *some* manager to keep your actual documents/Portable Document Formats (PDFs) and the automatically derived `*.bib` files in one place, named and structured consistently.

1.3 Lists

In technical publications, using lists (either bullet points or enumerations) is highly encouraged. They should *always* be preferred over doing the same thing in a block of text. Just consider this example list:

- The presented approach is more complex than the previous one:
  1. more time was spent doing computations,
  2. less was spent effing about,
  3. features were added.
- At the same time, the following simplifications were made:
  1. went from continuous- to discrete-time simulations,
  2. threw out some superfluous stuff.

Processing the very same information from a text block would suddenly be much more challenging. Note the block-like `itemize` symbol (▪)[2], and the fact that enumerate numbers are part of the sans-serif family and **bold**. That is totally cool and stuff, because it's... different? This was specified and may be changed in the **enumitem** package options.

1.4 Censoring

I am a TODO note from **todonotes**. I am also specially highlighted in many editors.

Let me tell you a huge secret: [redacted]. You can also censor float contents, as illustrated in Figure 1.1.

1.5 Glossaries

The package **glossaries-extra** is an absolute beast. For this document, it is used with its back-end **bib2gls**. This (Java) tool comes with installations of TeXLive and Miktex. You should already have it.

In a format similar to normal bibliographies, we now manage all abbreviations, symbols *etc.* using `*.bib` files. You can see them in a sub-folder for this project as well. Each entry is processed by **bib2gls** and put into an auxiliary file. From it, it reads and inserts all contents when we use `\gls` and its many siblings to insert glossary data. The framework for all of that, also printing of the glossary/nomenclature, was already done for this project.

Using **glossaries-extra** and its many features also for symbols is a fabulous tool. It is absolutely counter-intuitive, for now we have to write `\gls{temperature}` instead

[2] This is also a regular Unicode character. When copy-pasting old or outdated LaTeX documents, ever noticed all the funny business going on? A good example is copying German 'Umlauts' like Ä, Ö, Ü. They might turn into '¨A', since LaTeX only ever put two random dots over the regular ASCII letter A. With **unicode-math** (building onto **fontspec**) and LuaLaTeX, we are fully Unicode! Copy-paste anything correctly: $\delta\sigma \int_1^2 y$.

Figure 1.1 / Example for a censoring box.

of just `T`. This makes the source code obfuscated and hurts legibility. The advantages are probably worth it: never worry about consistency again. You have one central repository for all data, and any changes there are immediately forwarded to the entire document (provided you are persistent in using `\gls` *etc.*). We can then also go ahead and add a billion other features, like cross-referencing, hyperlinking, additional explanations and much more.

1.6 Indices

For the index at the end, we currently use **splitidx**. It splits the single index LaTeX expects, allowing for multiple indices. In the future, all of this should be replaced by **glossaries-extra**. For now, we can add people like Einstein and Feynman to the index. Terms, like machine learning and artificial intelligence (so hip), may be added as well. We can also nest entries with `\term{parent!child}` .

These are implemented as a custom commands and are vastly inferior to using **glossaries-extra**. For example, these are simple links with no descriptive texts *etc.*

Lastly, we will have to run the tool `splitidx` on our main `*.tex` file/its `*.aux` file. This is another step in the process that could be circumvented using **glossaries-extra**.

1.7 Code

As previously seen in Section 1.1.3, we have a special code-block environment for code between paragraphs (*i.e.*, not a float). Its big brother Floaty McFloatface is illustrated in algorithm 1.1. Notice how you can very easily select and copy[3] said code without

[3] steal

Code 1.1 / This is an example for a floating code environment with some interesting LaTeX stuff. The actual code is likely crap, you be the judge.

```
def value_cleanup(raw_in) -> float:
   """Turn dirtied string(s) (e.g. ",233 kg") to float(s)."""
   if isinstance(raw_in, (int, float, datetime)) or raw_in is None:
      return raw_in A useless note: x^2 ≠ x
   elif isinstance(raw_in, list):
      clean_list = []
      for substr in raw_in:
         clean_list.append(value_cleanup(substr))  # Recursion
      return clean_list
   elif not isinstance(raw_in, str):
      raise TypeError(f"Expected type 'str', got '{type(raw_in).__name__}'.")
   else:
      dotted = raw_in.replace(",", ".")  # Decimal representation
      cleaned = dotted.strip()  # Remove surrounding whitespace
      numeric = '0123456789-.'  # Include negatives/decimal sep. in search
      position = None  # Initialize to throw error just in case
      # Append space for search to work
      for position, char in enumerate(cleaned + " "):
         if char not in numeric:
         if position == 0:  # Didn't even start with numerical char
            return None This is interesting
         break
      # Up to just before found nun-numeric char
      return float(cleaned[:position]) -1.5 × 10^-2 is also a float
```

having to worry about line numbers. The line numbers can even be referenced, *e.g.* we find a `return` statement on Line 4. This requires escaping the `\label` with the previously specified special command, back-ticks: `` `escaped LaTeX` ``.

Inline code goes like so: `y = [file_patterns[x] for x in ["send", "help"]]`.

1.7.1 Matlab/Simulink

The basis for this template was laid by a thesis concerned with MATLAB®/Simulink®. As such, we have nice commands like MATLAB®, Simulink®, MATLAB®/Simulink® and Symbolic Math Toolbox™ to print, in a consistent manner, all the different names and also add them to the index. Further, there are funny little vector symbols available, referencing various things MATLAB®/Simulink® has to offer:

Inputs: *Step-Function* *Step-Functions*
Data Dictionaries: *Data Dictionary* *Data Dictionaries*
Workspaces: *Base Workspace* *Model Workspace* *Model Workspaces*
Models: *Model* *Models* *Referenced Model* *Referenced Models* *Library Model* *Library Models* *Library* *Libraries*
Data processing: *Data Logging*
Look-Up Tables: *Look-Up Table* *Look-Up Tables*
Settings: *Configuration*
Variables: `YourVar` *Struct* *Structs* *Table* *Tables* `YourStruct` `YourObj` *Bus* *Buses*

These can probably be improved by handling plural cases/commands automatically. Since plural forms and all that jazz are often not as straightforward as appending an ‘s’, and regarding the countless exceptions one would have to take into account, automating LaTeX macros that deal with language is a much harder task than what it might seem at first.

1.8 Landscape

Pages in landscape format are rather straightforward to implement. Note that not only are these in landscape orientation; they are also recognized as such by supporting PDF viewers, rotating the page for you and keeping it legible.

Chapter 2

Float Features

In Figure 1.1 and Table 1.1, we have already seen examples for floats. These are handled by packages **caption** and **floatrow**. A notable feature is the capability for captions the same width as the float they are attached to. This tends to look much prettier, *cf.* Figures 2.1 and 2.2.

2.1 Multiple floats

Other possibilities are rather arbitrary arrangements of sub-figures and -captions. For this, see Figure 2.3, which contains two sub-figures Figures 2.3a and 2.3b. Multiple sub-tables are also possible, see Table 2.1.

2.2 Side-Captions

Lastly, on occasion figures and their captions might look disproportionate in combination. In these cases, placing a side-caption might relieve the situation, as shown in Figure 2.4.

2.3 Caption Positioning

Note that the caption of Table 1.1 occurs *above* the table no matter the **caption** command's position. Per convention, figure captions should appear below, table captions above their bodies. This is also handled by **floatrow**. Also note that there is neither a fullstop nor *any* character (no space, no empty line) behind the last caption line in the source code, since dots are managed globally by the **caption** package.

Float Footer We also have a `\floatfoot` command for all floats. This is used to place additional info underneath the caption, primarily used for references, *cf.* Figure 2.3.

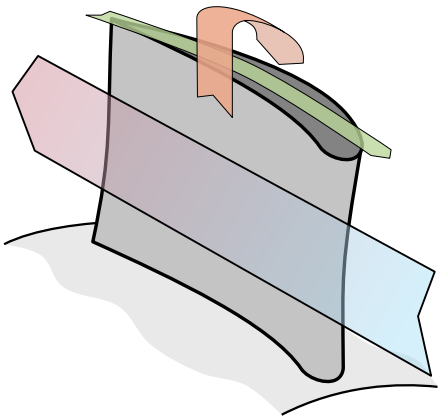


Figure 2.1 / Example for a regular caption, spanning the whole width since it is so long.

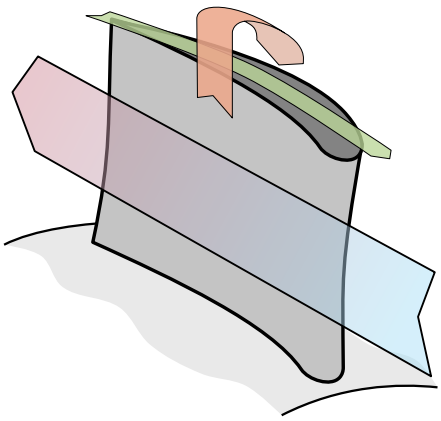
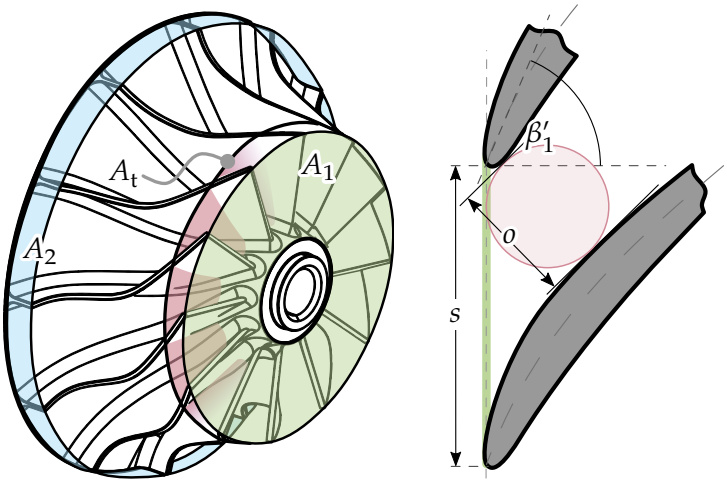


Figure 2.2 / Example for a new caption, spanning the just the width of the float it is attached to.



a / Isometric. **b /** Schematic.

Figure 2.3 / Inducer throat created by geometric constraints as an example for sub-figures.

Adapted from (Shakal 2015; Whitfield and Baines 1990, p. 99; Hayami, Senoo, and Ueki 1985, p. 535).

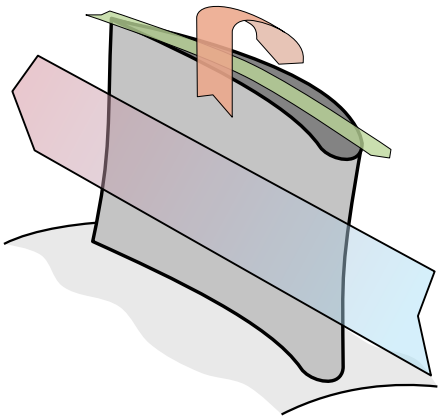


Figure 2.4 / A side caption, which may also span multiple lines like demonstrated in this rather long caption right here.

Table 2.1 / Works on zero- and one-dimensional turbocharger and engine modelling.

a / Using Maps.

Author & [Work]	Comment
Bozza and De Bellis (2011)	—
Bozza, De Bellis, et al. (2013)	—
Bozza, Nocera, et al. (1990)	**Nozzles**
Burke (2014)	—
De Bellis and Bontempo (2018)	—
Berndt (2009)	—
Chesse et al. (2000)	—
Cornolti et al. (2013)	—
L. Eriksson (2007)	—
Grigoriadis (2008)	—
Kech and Klotz (2002)	—
Lee, Filipi, et al. (2009)	—
Leufvén and Lars Eriksson (2011)	—
S. Shaaban et al. (2006)	—
Wahlström and L. Eriksson (2011)	—
Watel et al. (2010)	MATLAB®/Simulink®
Yang and Zhu (2010)	—

b / Computing Maps.

Author & [Work]	Comment
Beckey, Hartfield, and Carpenter (2011)	using algorithms
Bergqvist (2014)	from CFD
Bohn et al. (2002)	—
Bolz (2014)	—
Elkamel et al. (2011)	—
Ewert (2013)	—
Freeman (2011)	MATLAB® script
Hänsch (2010)	—
Harley, Spence, et al. (2014)	—
Lüddecke (2014)	—
Kurzke (2011)	—
Schwarz (2014)	—

c / Physical (0- & 1D).

Author & [Work]	Comment
R. H. Aungier (1995)	—
Casey and Robinson (2013)	Calibrating with Maps
H. Chen and Winterbone (2014)	Turbine
Erickson (2008)	FORTRAN
Ewert (2013)	—
Gong and R. Chen (2014)	—
Gutiérrez Velásquez et al. (2010)	FORTRAN, 1D and 3D
David Japikse (2009)	Overview
Kamaleshaiah, Venkatrayulu, and Ramamurthy (1988)	—
Kessel (2002)	1- & 3D
Lee, Jung, et al. (2008)	dual-stage; **Simulink®, maps**
Nakhjiri, P. F. Pelz, et al. (2018)	—
Okhuahesogie et al. (2014)	Differential Evolution
Pelton (2007)	—
Pixberg (2013)	**Reservoirs; Nozzles** (p. 42)
Sakellaridis et al. (2015)	—
Schiffmann and Favrat (2010)	—
Schur (2013)	Simulink®
Serrano et al. (2008)	**Reservoirs; Nozzles**
Sameh Shaaban (2004)	Diabatic; Extrapolation
Stuart et al. (2014)	cites (Harley, Early, et al. 2013)
Taburri et al. (2012)	**Reservoirs; Nozzles**
Uchida (2006)	—
Whitfield (1990)	Preliminary Design
Xu and Amano (2012)	—
Zahn (2012)	—
Zhuge et al. (2009)	—

d / Loss Modelling.

Author & [Work]	Comment
R. H. Aungier (1995)	—
Doustmohammadi, Hajilouy Benisi, and Mojaddam (2013)	—
El-Maksoud, Bayomi, and Rezk (2012)	—
Galvas (1972, 1973)	—
Gülich (2003)	Disk Friction
Gutiérrez Velásquez (2017)	—
Harley, Early, et al. (2013)	Recommends (Oh, Yoon, and Chung 1997)
Mohtar (2010)	—
Nakhjiri, P. Pelz, et al. (2011)	—
Oh, Yoon, and Chung (1997)	—
Tacconi (2018)	Cites (Harley, Early, et al. 2013); recommends (Galvas 1973)
Schneider et al. (2015)	2D
Solaesa (2016)	—

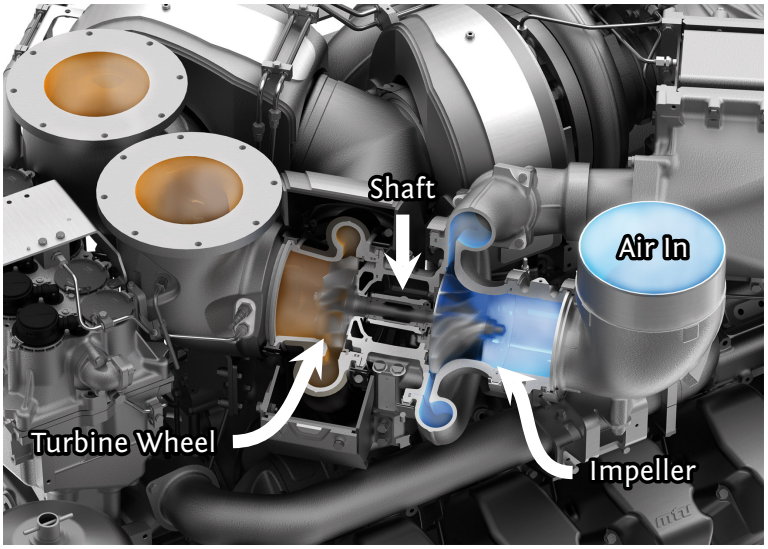


Figure 2.5 / Digital renders have to be implemented as bitmaps. You can actually vectorize images like this one, too, with decent success. However, the resulting file will contain an absurd number of paths and can be multiple megabytes in size. We can draw onto bitmaps using Tikz. .

Adapted from (Rolls-Royce Power Systems AG 2011). In the current `*.bib` file, this reference was automatically generated by Zotero as `@artwork`. While that is okay, `biblatex` will throw a warning that there is no driver for such a type. It falls back to `@misc`, which works just fine.

2.4 Tikz and pgfplots

Packages `tikz` and `pgfplots` offer an absolutely scary plethora of features. A select few are presented here.

2.4.1 Drawing over Bitmaps

When having to rely on bitmaps, once might still want to add additional info. This can be done directly in LaTeX, profiting from all the usual features. In the example here, this is of course the retaining of the text font, but also usage of the wonderful `contour` package to draw legible black-on-white (or vice-versa) text. An example is shown in Figure 2.5.

2.4.2 Direct plotting

If you rely on tools like `matlab2tikz`, maybe this is for you. We can plot *directly* into LaTeX, without having to import outside data in the form of `*.csv`-files or automatically generated Tikz-pictures. Still, anything more than polynomials is probably still too much. While the functionality is limited, it may still save a lot of time and headaches.

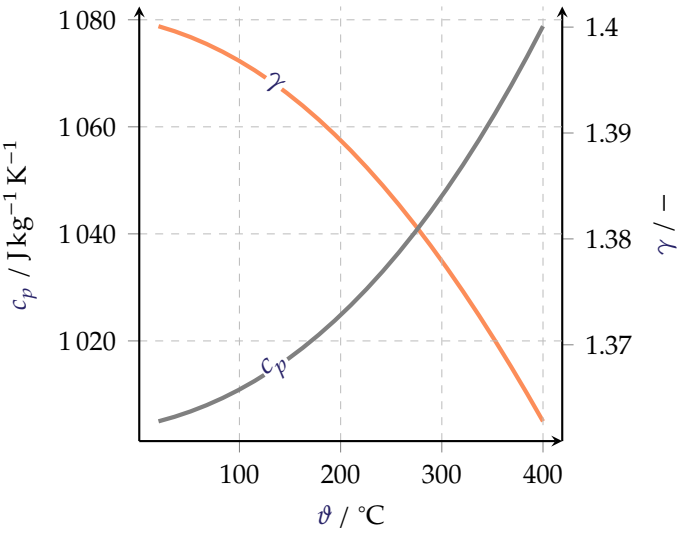


Figure 2.6a / Caloric parameters of air. **Avoid legends** and put info where it belongs, improving legibility (less back-and-forth for the eye).

See (Dixon and Hall 2014, p. 15).

This is demonstrated in Figures 2.6a and 2.6b.

2.4.3 Plotting from files

As discussed, often we'd want to plot data from files. The better behaved the CSV file is (meaningful headers, no junk rows), the easier that is. In Figure 2.7, we only have to specify *e.g.* `y=M` and the column corresponding to that header is automatically chosen, with no confusion about indices/numbers.

2.4.4 Tikz and Text

We can also draw tikz content into text content using `\tikzmark`. This, and also how to use `\foreach` in tikz, is illustrated in Equation 2.1. There, usage of chemical compounds as `\chcpd` is also shown.

$$\gamma_{\mathrm{C}} + \gamma_{\mathrm{H}} + \gamma_{\mathrm{S}} + \gamma_{\mathrm{O}} + \gamma_{\mathrm{N}} + \gamma_{\mathrm{H_2O}} + \gamma_{\mathrm{ash}} := 1\,. \qquad [2.1]$$

Carbon, Hydrogen, Sulphur, Oxygen, Nitrogen, Water, Ash

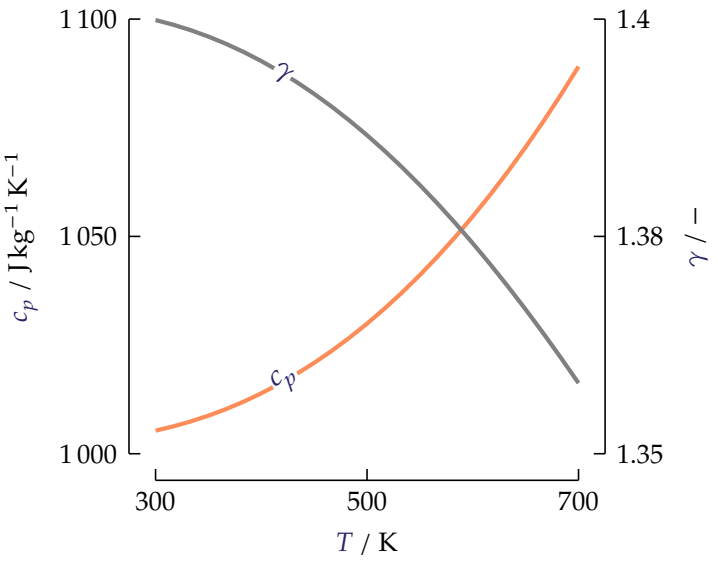


Figure 2.6b / Same as Figure 2.6a in hip and '*Tufte*-like'. I think it looks gorgeous. For more info, refer to its namesake, Tufte. Note how this is a *continued* float.

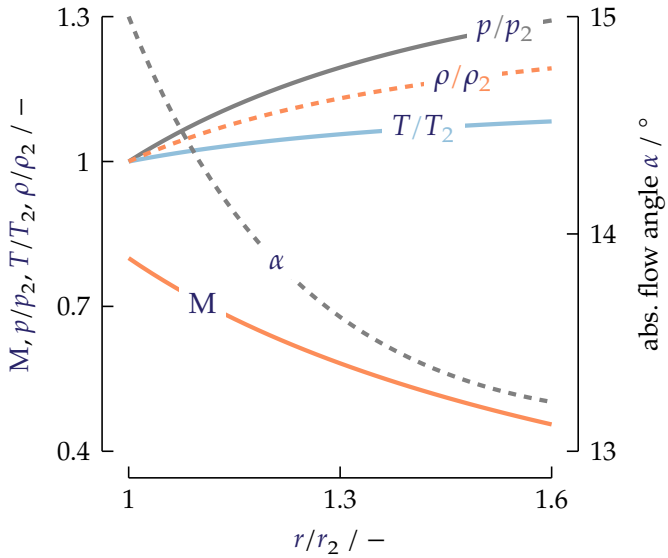


Figure 2.7 / A plot from CSV data for a diffuser.

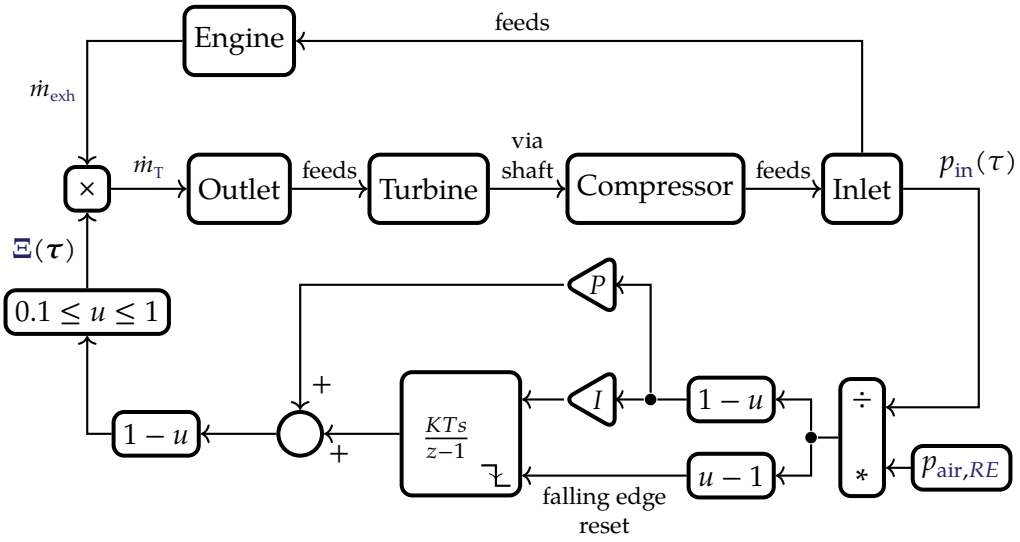
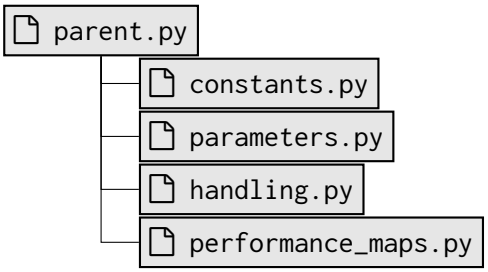


Figure 2.8 / Wastegate implementation in a feedback-loop in Simulink® as an example for a Tikz diagram.

Tikz are much improved, for it is suited well then. For example, a small file structure diagram:



Note how `tikzpicture` environments don't have to be contained in floats. A second example is shown in Figure 2.8.

2.4.6 InkScape

Having seen what Tikz is good for, Figures 2.9a and 2.9b are good examples for when InkScape might be the better choice: 3D drawings with curves. Figure 2.3a is a bitmap turned vector; InkScape can detect edges/contrasts in bitmaps and replicate those lines in a vector graphic. While InkScape is really awkward to use at times, and severely lacking behind its commercial competitors (which is fine), it has 3D perspective tools that can also handle these cases. That being said, Tikz is always the preferred method if so feasible.

InkScape will let you create freely scalable vector graphics (`*.svg`). They may then be exported as `*.pdf`; now, that is only half the story. The vector graphic's text part is

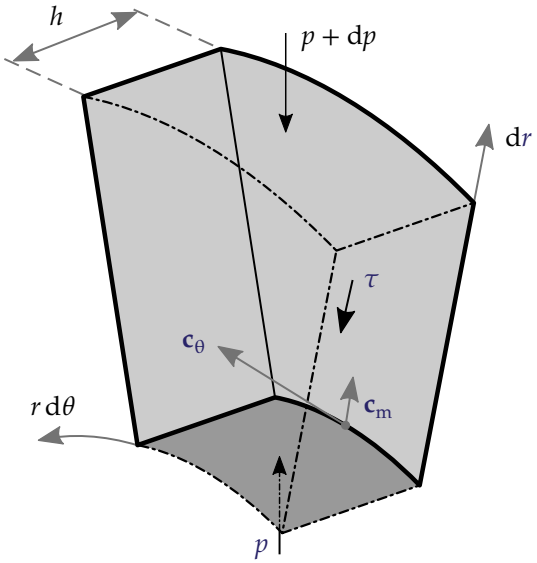


Figure 2.9a / InkScape is likely a better choice for images like this one.

Adapted from (Stanitz 1952, p. 47).

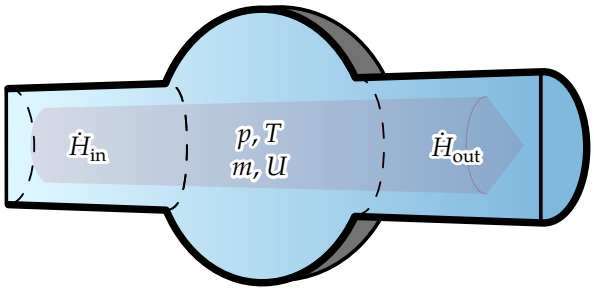


Figure 2.9b / Another example for InkScape usage.

Adapted from (Guzzella and Onder 2010, p. 30).

exported to a plain text (*i.e.* TEX) file, as `*.pdf_tex`. On the latter, we then call `\input` within LATEX, and that file in turn embeds the previously created `*.pdf` file.

Single-Page PDFs If you aim to reuse just the `*.pdf`, without the text, outside of LATEX, perhaps on your website, then you will want it to really only contain *one* page. However, InkScape may create multi-page files. Selecting all text elements in the `*.svg` file and moving them to the top-most layer (press `POS1`) will ensure that the generated `*.pdf` will have just one page containing the entire image.

2.5 Illustrations

As a special gimmick, there is an environment for illustrations. It may be useless now, but you can alter it to suit your needs; the skeleton is there for you. It mainly behaves like the other floats: it...floats (if you allow it to) and also has its own list. For an

example, see Section 2.5.

Illustration 2.1: I am a useless box

There can be pretty much any content in here. Math works, as we can see

$$1 = 1! \qquad [2.2]$$

still holds true after all these years. Inserting other floats in here will cause LaTeX to have a massive fit. We circumvent this by setting the `[H]` flag, saying 'Oh hell yeah, we do want this float right here'. **In any other context, setting any such flag is considered poor style.** Well, by me at least. You **will** be setting these flags prematurely and then run into countless issues of placement and **very poor** spacing. For the love of God, let LaTeX do its job in placing the floats. Truth be told, they will on occasion not be placed where you need or want them. Keep working. At the very end, when all is done, go ahead and change the few misplaced floats manually, by shoving them about in the source code (still not using `htb!` flags). Then, you're done, with minimal pain and maximum usage of LaTeX spacing/placing capabilities. Here is such a float within a float:

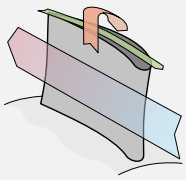


Figure 2.10 / Side-captions are still possible. So are labels.

This box even breaks across pages if so required. Should this turn out ugly, some manual action is certainly required.

Chapter 3

Style

These are just random thoughts on style and design of documents, primarily technical publications of all kinds. They are a collection of experience, but also aspects taken from various literature on the topic. The available literature is positively exhausting, so there is no point digging into that here. Take to it if you are truly interested; the content presented here is on a beginner level.

Of course, anything is subjective here. When one speaks of 'rules' in style and design, of course what they are really saying is '99 % of people agree on this'. Many people make an effort to be the other 1 % (guilty!). Chances are, it will not come out looking right and you'll be back to the drawing board anyway.

3.1 Conventions

Less is more: hack and alter as little as possible. Armies of professionals have arrived at today's conventions over centuries. One example that breaks with what is usually found in documents are the page headers of this document. The header height is increased beyond its standard (using the fabulous `koma-script`) and a vertical rule was inserted. I like it: a touch different and modern, yet not in the way of anything. If you don't, feel free to tear it down in the preamble.

Less is more: use the least ink possible to get your point across. Any more is only noise. For this, compare Table 1.1 with its not-so-blessed twin, Table 3.1. Another observation there: **if in doubt, left-align**. If you don't have an *actual reason* to center or right-align, just don't.

An example for a larger table is shown in Table 3.2. One key aspect there: column-types provided by `siunitx`. These automatically apply `\num` to each entry, which in turn allows easy printing of things like '-3.23×10^{-5}'. Use `*x{y}` to print column-type `y` (*i.e.* `c`) `x` times. No need for tedious repetition. Decimal places are accounted for and aligned by.

Table 3.1 / Awful version of Table 1.1.

Feature	*Sample Text*
Regular	The quick brown Fox jumps over the lazy Dog 13 times!
Bold	**The quick brown Fox jumps over the lazy Dog 13 times!**
Italics	*The quick brown Fox jumps over the lazy Dog 13 times!*
Bold Italics	***The quick brown Fox jumps over the lazy Dog 13 times!***
Small Capitals	The quick brown Fox jumps over the lazy Dog 13 times!
Bold SC	**The quick brown Fox jumps over the lazy Dog 13 times!**
Italics SC	*The quick brown Fox jumps over the lazy Dog 13 times!*

Table 3.2 / Compositions and properties of fuels, as used in Eq. 2.1.

Name	Mass fraction γ [–]							ρ $[\mathrm{kg\,m^{-3}}]$	H_i $[\mathrm{MJ\,kg^{-1}}]$
	C	H	S	O	N	H_2O	Ash		
Diesel[a]	0.8600	0.1320	0.0060	0.0020[b]		n.a.	n.a.	840	42.7
Oil EL[a]	0.8570	0.1310	0.0100	0.0020[b]		n.a.	n.a.	840	42.7
Oil H[a]	0.8490	0.1060	0.0350	0.0100[b]		n.a.	n.a.	980	40.0
MDO[c]	n.a.	n.a.	0.0150	n.a.	n.a.	0.0030[d]	0.0001	900	n.a.
HFO[e]	n.a.	n.a.	0.0350[f]	n.a.	n.a.	0.0050[d]	0.0015	1010	n.a.
Light[g]	0.8600	0.1320	0.0060	0.0010	0.0010	0	0	840	Eq. 1.7
Medium[g]	0.8530	0.1269	0.0150	0.0010	0.0010	0.0030	0.0001	900	Eq. 1.7
Heavy[g]	0.8460	0.1025	0.0350	0.0050	0.0050	0.0050	0.0015	1010	Eq. 1.7

[a] (Baehr and Kabelac 2016, p. 634); see also (Dubbel, Grote, and Feldhusen 2007, p. L70) and (Mollenhauer and Tschöke 2007, p. 97).
[b] Given as a sum $\gamma_\mathrm{O} + \gamma_\mathrm{N}$.
[c] MDO, max. values of specification *DMB*, (International Organization for Standardization 2017).
[d] Given as a volume fraction, assumed equal to mass fraction.
[e] HFO, max. values of specification *RMK*, (International Organization for Standardization 2017).
[f] IMO level prior to 2020.
[g] Derived, *virtual* fuels for simulation.

3.2 Sectioning

3.2.1 Explanation

Between any two sectional commands (from `\part` down), there always has to be *some* content. Anything else looks off, *cf.* Sections 3.2 and 3.2.1. At least, tell the reader what the following hierarchical level contains for them.

Further, three numbered levels of hierarchy are enough. In **koma-script** reports (`scrreprt`), these are chapters, sections and subsections.

Deeper!

Deeper sectioning is fine, but it should not be numbered or occur in the table of contents. Notice how, without specifying anything, **koma-script** abides by that automatically. This is despite using `\subsubsection{Deeper!}` as opposed to `\subsubsection*`, its explicitly unnumbered counterpart.

Also notice how between `\section{Sectioning}`, which produced Section 3.2, and `\subsubsection{Deeper!}`, there was no sub-section. Usually, you would want a clean cascade with no jump in the hierarchy levels. If you catch yourself jumping, it is also a sign there is something off with your actual content structure and thought process. That being said, I sometimes still do it because of reasons. Your call, just be aware.

Amazingly, LaTeX has many ways where it does not let you do something easily, or where commands look off. **Never fight LaTeX** in that; you will lose, it will win. When something is awkward to do or requires a lot of manual labor, you are probably doing it wrong and there will be an easier way. Most often, that comes in the form of packages. Being a fully-fleshed programming language under the hood, *any* repetition in LaTeX should make you stop for a second and wonder about another way. That may lead to you discovering sub-optimal approaches and code. Just as often, it won't lead anywhere and trying to force minimum code and maximum optimization will not be economical.

Paragraphs They are a nice touch, introducing a new, distinct paragraph/idea/-thought without being too loud about it.

That being said, leave a blank line in the source code for regular paragraphs. Use either vertical spacing between paragraphs or indentation (like here). Do not use both.

Each new train of thought should go into its own paragraph. This paragraph is not indented, as was intended by manually calling `\noindent`. In regular text and documents, there are very few reasons that should ever occur. Again, let LaTeX do its thing. Again, if you find yourself typing `\noindent` all the time, there will be something wrong.

3.3 Other stuff

This section is about auxiliary stuff. Notice `\lcnamecref` in the previous sentence: it references (in *l*ower-*c*ase) the name of the thing you are passing it. If this section ever changes to something else (chapter, ...), it is updated automatically.

Now for the stuff:

- Small spaces are produced by the usual back-slash, followed by an actual space: `\ `. Much like physical quantities produced by **`siunitx`** ($30.2 \times 10^2\,\mathrm{N}$) and its small spaces, they are nice to have. When LaTeX sees any dot `.` with a space behind it, it sees that as the end of the sentence. When it isn't, like for Prof. Very Important, we use an explicit small space instead.
  A second important application would be abbreviations, *i.e.* ← these little things. **In the name of all that is holy, never type units, numbers or quantities out by yourself, manually.** Use **`siunitx`** for that. It will insert a small space between the number and the unit. No spaces or full spaces are wrong and painful to read. Lastly, even numbers should always go into `\num`. Compare 47 sheep versus 47 sheep. Different fonts are used: the regular text font, for which we specified `OldStyle`, *i.e.* hanging numbers, the the math font. Other features are global control over decimal and group/thousands separators, how to handle scientific notation (1×10^1) and *much* more. As always, the manual will be your reference and friend.
- Use `---` in the source code to have an em-dash–perhaps even without surrounding spaces? That is probably up to your choice. What is not, because it is largely agreed on, is to use `--` – an en-dash – instead. It is reserved for number ranges. Lastly, a single hy-phen is reserved for just that — hyphenation.
- A generally important concept in LaTeX: **write what you *mean*, not what produces the desired output**. Do *emphasized* text and *emphasized*, but in italics, text look the same? They certainly do, usually.
  Yet, what did we *mean* to be doing? We *mean* to *emphasize*; italic text is just what it happens to look like now, but it is not the *meaning*. For example, we could later decide to redefine emphasized text to bold. If you previously did not differentiate strictly enough between `\emph` and `\textit`, you are in for a bad time.
- Indices are upright! They are text. If text occurs in math mode, that is also actual text. Consider

$$MEAN_{sample} \neq 2\,, \tag{3.1}$$

  which looks stupid. What Equation 3.1 really says is '$M \cdot E \cdot A \cdot N$': slanted characters are variables, and not specifying any operator implies multiplication. This example might be pedantic, but it is easy to imagine such an issue evolving into real ambiguity, which would be a serious error. Since multiplication is implied by doing nothing, leaving out `*` aka `\cdot` altogether often looks best.
  We can `\DeclareMathOperator{\examplemean}{MEAN}`, then use `\text` for the subscript:

$$\mathrm{MEAN}_{\mathrm{sample}} \neq 2\,. \tag{3.2}$$

- In Equation 3.2, and anywhere else, the equation is part of the sentence. It is just another language, which can and will be read as a natural part of the surrounding text. As such, it should contain punctuation marks. So that now, having considered that we have the amazing result of

$$1 \neq 2\,, \tag{3.3}$$

  we have achieved better overall reading flow. Notice the small spaces `\,` before the commas and dots in equations.
  Implementing these as `\eqcomma` and `\eqend` ensures consistency. Note that `\eqend` was previously `\eqdot`, but that is reserved by **`unicode-math`**. Other than consistency, the punctuation can then also be switched off globally if your requirements demand so.

Chapter 4

Technical Aspects

TEX being TEX, there are countless of technical aspects to pay attention to. The most important one are collected here, in a random fashion.

4.1 Building the document

The most important part! In Chapter 1, we looked into why we are using LuaLATEX. Reasons are:

1. `unicode-math` aka `fontspec`
2. `contour` works (it does not with XELATEX)
3. it allocates as much memory as it happens to need. The ancient limitations of TEX are easily reached with `tikz` and `pgfplots` functionalities. Circumventing that either feels hacky or actually is hacky. `tikzexternalize` is a decent module, but has a long list of caveats. Using LuaLATEX, we do not have to worry about any of that: hit compile, wait, done (the waiting part might be a bit longer than what you might be used to from pdfLATEX).

With that out of the way, the building process — starting from scratch — looks roughly like this:

1. **LuaLATEX** — build first version, generating the auxiliary files. At this point, references or anything else will not yet be correct.
2. **biber** — it will take the specified `*.bib` file and build the bibliography auxiliary files.
3. **splitindex** — invoke it as `splitindex NAME`, where NAME is your main file's name, such that all the auxiliary files bear names like `name.aux` *etc.*
4. **bib2gls** — also invoked as `bib2gls NAME`,
5. **LuaLATEX**,
6. **LuaLATEX**.

This list is no guarantee however. If anything fails, try running it again in different orders, or run LuaLATEX again. You can also look into deleting auxiliary files (they can always be regenerated freely) if you are seeing really cryptic error messages.

4.1.1 latexmk

A much more user-friendly tool is `latexmk`, also provided in most TEX distributions. Therefore, you should already have it available and on your PATH.

The tool knows about all the different auxiliary files and will run your compiler of choice as many times as needed. It will know when to stop by checking the auxiliary files: if they do not change anymore between runs, we must have hit a 'steady state' and the produced document will be ready.

Since it does not natively do well with either **`glossaries-extra`** or **`splitidx`**, this project contains a file `latexmkrc` in its root. It is a configuration file written in Perl, letting `latexmk` know about the auxiliary files of those two packages and what to do with them. The tool will look for its configuration file in the project root and find it there.

Now, you can start from absolute scratch, with no auxiliary files generated yet at all. All you need is a TEX distribution and Java (for **`bib2gls`**)! I recommend TeXLive: in its full installation, it ships with all the required tools.

Then invoking **`latexmk yourfile`** will run the entire pipeline, giving you the fully-done end product.

4.1.2 Version Control

LATEX is well suited for a version control system, specifically Git. Using GitHub or GitLab, you can also have your document built automatically, on servers specified (and perhaps paid for) by you. You could be editing LATEX source code on your phone, push it to the remote repository, and download the compiled document.

If using Git, each text line should also be on its own line of code. Should you bunch multiple text sentences into one line of code, diffing will be harder.

Continuous Integration

If you are on GitLab, you can configure your runner with the YAML file found in the project root (the runner will expect it there anyway). The runner you need to provide yourself; perhaps, your university has one. I do not yet have an analogous Travis-CI configuration for GitHub.

Docker The Continuous Integration (CI) configuration for GitLab relies on a docker image found here:

`https://cloud.docker.com/u/alexpovel/repository/docker/alexpovel/javalatex`

Its `Dockerfile` does `apt-get update -y texlive-full` and as such, will download a *massive*, full TeXLive distribution. That is still worth it and nice to have, so we do not have to specify packages manually ourselves. The file's second important part is of course Java for **`bib2gls`**.

Metadata In the project root, you will notice a file `gitmeta.txt`. This is my pathetic approach of breathing some dynamic metadata printing into the produced document. You can see the result of that on Page b of this document, right after the title page. LaTeX is made to read the contents of that `gitmeta.txt` file and implements it. The text file's default contents are printed if we do not involve it in our building process.

We could change the `*.tex` contents directly using the `sed` commands from the `.gitlab-ci.yml` file, but I reckoned going for an external file would make the `*.tex` file more robust and stand-alone.

You can will the file out manually (highly discouraged and besides the point) or have it, as is configured for GitLab in `.gitlab-ci.yml`, done by the building process on your runner.

That way, you will never lose track of your finished versions either. Usually, you won't be able to tell what commit or tag a generated PDF belongs to once it is in circulation. But hard-printing that, we can always go back to the very specific commit it belongs to.

4.2 Random Thoughts

- If you don't intend to allow `\par` in your custom macros, use the starred variant of `\newcommand`.
- Put `{braces}` around anything where it is applicable. That includes sub- and superscripts that consist of only one character. `x^2` will work, but `x^{2}` will ensure nothing breaks down the line, *e.g.* when automatically replacing stuff. Superfluous braces don't hurt, but their lack can bite you.

Rest of this page intentionally left blank.

Bibliography

Aungier, R. H. (1995). 'Mean Streamline Aerodynamic Performance Analysis of Centrifugal Compressors'. In: *Journal of Turbomachinery* 117.3, pp. 360–366 (cit. on p. 14).

Aungier, Ronald H. (2000). *Centrifugal compressors: a strategy for aerodynamic design and analysis*. New York: ASME Press. 315 pp. (cit. on p. 4).

Baehr, Hans Dieter and Stephan Kabelac (2016). *Thermodynamik: Grundlagen und technische Anwendungen*. 16., aktualisierte Auflage. Lehrbuch. Berlin: Springer Vieweg. 671 pp. (cit. on p. 23).

Beckey, Capt Carl, Roy Hartfield, and Mark Carpenter (July 2011). *Compressor Modeling for Engine Control and Maintenance*. AFFTC-PA-11253. Edwards Air Force Base, California, USA: Air Force Flight Test Center, p. 16 (cit. on p. 14).

Bergqvist, Stephanie (2014). 'Prediction of Turbo Compressor Maps using CFD'. Master Thesis. Gothenburg, Sweden: Chalmers University of Technology (cit. on p. 14).

Berndt, René (Oct. 2009). 'Einfluss eines diabaten Turboladermodells auf die Gesamt-Prozess-Simulation abgasturboaufgeladener PKW-Dieselmotoren'. PhD thesis. Berlin: Technische Universität Berlin. 131 pp. (cit. on p. 14).

Bohn, D. et al. (2002). *Modellierung des Wärmeflusses im und am System Turbolader*. Abschlussbericht Vorhaben Nr. 755. Aachen: Institut für Dampf- und Gasturbinen, RWTH Aachen, p. 101 (cit. on p. 14).

Bolz, H (2014). 'Critical aspects in turbocharger testing'. In: p. 13 (cit. on p. 14).

Bozza, Fabio and Vincenzo De Bellis (Sept. 11, 2011). 'Map-Based and 1D Simulation of a Turbocharger Compressor in Surging Operation'. In: *SAE International Journal of Engines* 4.2, pp. 2418–2433 (cit. on p. 14).

Bozza, Fabio, Vincenzo De Bellis, et al. (Nov. 2013). 'Numerical analysis of the transient operation of a turbocharged diesel engine including the compressor surge'. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 227.11, pp. 1503–1517 (cit. on p. 14).

Bozza, Fabio, Roberto Nocera, et al. (Sept. 1, 1990). 'Theoretical and Experimental Investigation of the Matching Between an I.C.E. and a Turbocharger'. In: International Off-Highway & Powerplant Congress & Exposition. Milwaukee, Wisconsin: SAE International (cit. on p. 14).

Bräunling, Willy J. G. (2015). *Flugzeugtriebwerke: Grundlagen, Aero-Thermodynamik, ideale und reale Kreisprozesse, thermische Turbomaschinen, Komponenten, Emissionen und Systeme*. 4th ed. VDI-Buch. Berlin: Springer Vieweg. 1997 pp. (cit. on p. 5).

Burke, R. D. (2014). 'Modelling of turbocharger heat transfer under stationary and transient engine operating conditions'. In: p. 10 (cit. on p. 14).

Casey, Michael and Chris Robinson (Mar. 2013). 'A Method to Estimate the Performance Map of a Centrifugal Compressor Stage'. In: *Journal of Turbomachinery*. 21034th ser. 135.2, pp. 1–10 (cit. on p. 14).

Chen, H. and D.E. Winterbone (May 2014). 'A one-dimensional performance model for turbocharger turbine under pulsating inlet condition'. In: *Conference Proceedings*. 11th International Conference on Turbochargers and Turbocharging. Woodhead Publishing, pp. 113–123 (cit. on p. 14).

Chesse, Pascal et al. (2000). 'Performance Simulation of Sequentially Turbocharged Marine Diesel Engines With Applications to Compressor Surge'. In: *Journal of Engineering for Gas Turbines and Power* 122.4, pp. 562–569 (cit. on p. 14).

Cornolti, L. et al. (Nov. 2013). '1D simulation of a turbocharged Diesel engine with comparison of short and long EGR route solutions'. In: *Applied Energy* 111, pp. 1–15 (cit. on p. 14).

De Bellis, Vincenzo and Rodolfo Bontempo (Jan. 2018). 'Development and validation of a 1D model for turbocharger compressors under deep-surge operation'. In: *Energy* 142, pp. 507–517 (cit. on p. 14).

Dixon, S. L. and C. A. Hall (2014). *Fluid mechanics and thermodynamics of turbomachinery*. 7th ed. Amsterdam ; Boston: Butterworth-Heinemann. 537 pp. (cit. on pp. 4, 5, 16).

Doustmohammadi, Ali Asghar, Ali Hajilouy Benisi, and Mohammad Mojaddam (June 3, 2013). 'Experimental and Numerical Investigation of Losses in Centrifugal Compressor Components'. In: *Volume 6C: Turbomachinery*. ASME Turbo Expo 2013: Turbine Technical Conference and Exposition. San Antonio, Texas, USA: ASME, V06CT40A015 (cit. on p. 14).

Dubbel, Heinrich, Karl-Heinrich Grote, and J. Feldhusen (2007). *Taschenbuch für den Maschinenbau*. 22nd ed. Berlin Heidelberg New York: Springer. 1 p. (cit. on p. 23).

Elkamel, Chiheb et al. (June 2011). 'Experimental study of the surge behaviour of a centrifugal compressor'. In: *EAEC 2011 13th European Automotive Congress*. Valencia, Spain: Polytechnique Valencia. (cit. on p. 14).

Erickson, Christopher Erik (2008). 'Centrifugal compressor modelling development and validation for a turbocharger component matching system'. Master Thesis. Manhattan, Kansas: Kansas State University. 75 pp. (cit. on p. 14).

Eriksson, L. (July 1, 2007). 'Modeling and Control of Turbocharged SI and DI Engines'. In: *Oil & Gas Science and Technology - Revue de l'IFP* 62.4, pp. 523–538 (cit. on p. 14).

Ewert, Masterarbeit Matthias (2013). 'Modellierung des Radialverdichters eines PKW-Abgasturboladers'. PhD thesis. Aachen: RWTH Aachen University (cit. on p. 14).

Freeman, Jeffrey Lee (Feb. 2011). 'Compressor Performance Map Generation and Testing per SAE J1723'. Bachelor Thesis. San Luis Obispo: California Polytechnic State University. 39 pp. (cit. on p. 14).

Galvas, M. R. (Mar. 1, 1972). *Analytical correlation of centrifugal compressor design geometry for maximum efficiency with specific speed*. Technical Report NASA-TN-D-6729. Washington, DC, United States: National Aeronautics and Space Administration, p. 37 (cit. on p. 14).

– (Nov. 1, 1973). *FORTRAN program for predicting off-design performance of centrifugal compressors*. Technical Report NASA-TN-D-7487. Washington, DC, United States: National Aeronautics and Space Administration, p. 57 (cit. on p. 14).
Gong, Xiaoyang and Rui Chen (2014). 'Total pressure loss mechanism of centrifugal compressors'. In: (cit. on p. 14).
Grigoriadis, Panagiotis (2008). 'Experimentelle Erfassung und Simulation instationärer Verdichterphänomene bei Turboladern von Fahrzeugmotoren'. PhD thesis. Berlin: Technischen Universität Berlin (cit. on p. 14).
Gülich, J. F. (Dec. 2003). 'Disk friction losses of closed turbomachine impellers'. In: *Forschung im Ingenieurwesen* 68.2, pp. 87–95 (cit. on p. 14).
Gutiérrez Velásquez, Elkin I. (Oct. 2017). 'Determination of a suitable set of loss models for centrifugal compressor performance prediction'. In: *Chinese Journal of Aeronautics* 30.5, pp. 1644–1650 (cit. on p. 14).
Gutiérrez Velásquez, Elkin I. et al. (2010). 'One and Three-Dimensional Analysis of Centrifugal Compressor for 600 kW Simple Cycle Gas Turbine Engine'. In: *Volume 5: Industrial and Cogeneration; Microturbines and Small Turbomachinery; Oil and Gas Applications; Wind Turbine Technology*. ASME Turbo Expo 2010: Power for Land, Sea, and Air. Glasgow, UK: ASME, pp. 471–476 (cit. on p. 14).
Guzzella, L. and Christopher H. Onder (2010). *Introduction to modeling and control of internal combustion engine systems*. 2nd ed. Berlin: Springer. 354 pp. (cit. on p. 19).
Hänsch, Susann (2010). 'Untersuchung des Einflusses geometrischer Parameter auf die Kennfeldbreite radialer Turboverdichter'. Diplomarbeit. Dresden: Technische Universität Dresden. 127 pp. (cit. on p. 14).
Harley, P. X. L., J. Early, et al. (2013). 'An evaluation of 1D loss model collections for the off-design performance prediction of automotive turbocharger compressors'. In: *IOP Conference Series: Materials Science and Engineering* 52.4, pp. 1–9 (cit. on p. 14).
Harley, P. X. L., S. W. T. Spence, et al. (2014). 'Inlet recirculation in automotive turbocharger centrifugal compressors'. In: p. 12 (cit. on p. 14).
Hayami, H., Y. Senoo, and H. Ueki (1985). 'Flow in the Inducer of a Centrifugal Compressor Measured With a Laser Velocimeter'. In: *Journal of Engineering for Gas Turbines and Power* 107.2, pp. 534–540 (cit. on p. 13).
International Organization for Standardization (Mar. 2017). *ISO 8217:2017 - Petroleum products - Fuels (class F) - Specifications of marine fuels*. 8217. File from www.danbunkering.com. ISO/TC 28/SC 4, p. 23 (cit. on p. 23).
Japikse, D. (Mar. 18, 1985). 'Assessment of Single- and Two-Zone Modeling of Centrifugal Compressors, Studies in Component Performance: Part 3'. In: *Volume 1: Aircraft Engine; Marine; Turbomachinery; Microturbines and Small Turbomachinery*. ASME 1985 International Gas Turbine Conference and Exhibit. Houston, Texas, USA: ASME, V001T03A023 (cit. on p. 4).
Japikse, David (Apr. 20, 2009). *Turbomachinery Performance Modeling*. SAE Technical Paper 2009-01-0307. Warrendale, PA: SAE International (cit. on p. 14).
Kamaleshaiah, M. S., N. Venkatrayulu, and S. Ramamurthy (1988). 'An Improved Method of Centrifugal Compressor Performance Prediction'. In: *Conference Proceedings*. 16th National Conference on FMFP, pp. 1–5 (cit. on p. 14).

Kech, J. M. and H. Klotz (Mar. 4, 2002). 'Model-Based Sequential Turbocharging Optimization for Series 8000 M70/M90 Engines'. In: SAE 2002 World Congress & Exhibition. Detroit, Michigan: SAE International (cit. on p. 14).

Kessel, Jens-Achim (Oct. 2002). 'Modellbildung von Abgasturboladern mit variabler Turbinengeometrie an schnellaufenden Dieselmotoren'. PhD thesis. Darmstadt: Technische Universität Darmstadt. 165 pp. (cit. on p. 14).

Kurzke, Joachim (2011). 'Correlations Hidden in Compressor Maps'. In: *Volume 1: Aircraft Engine; Ceramics; Coal, Biomass and Alternative Fuels; Wind Turbine Technology*. ASME 2011 Turbo Expo: Turbine Technical Conference and Exposition. Vancouver, British Columbia, Canada: ASME, pp. 161–170 (cit. on pp. 4, 14).

Lee, Byungchan, Zoran Filipi, et al. (Apr. 20, 2009). 'Simulation-based Assessment of Various Dual-Stage Boosting Systems in Terms of Performance and Fuel Economy Improvements'. In: *SAE International Journal of Engines* 2.1, pp. 1335–1346 (cit. on p. 14).

Lee, Byungchan, Dohoy Jung, et al. (2008). 'Dual-Stage Turbocharger Matching and Boost Control Options'. In: *ASME 2008 Internal Combustion Engine Division Spring Technical Conference*. ASME 2008 Internal Combustion Engine Division Spring Technical Conference. Chicago, Illinois, USA: ASME, pp. 267–277 (cit. on p. 14).

Leufvén, Oskar and Lars Eriksson (Jan. 1, 2011). 'Surge and Choke Capable Compressor Model'. In: *IFAC Proceedings Volumes*. 18th IFAC World Congress 44.1, pp. 10653–10658 (cit. on p. 14).

Lüddecke, B. (2014). 'Engine crank angle resolved turbocharger turbine performance measurements by contactless shaft torque detection'. In: p. 20 (cit. on p. 14).

El-Maksoud, Rafea M. Abd, Nazih. N. Bayomi, and Mohamed I. F. Rezk (Dec. 2012). 'Prediction and Modification of Centrifugal Compressor Performance Maps'. In: *Al-Azhar University Engineering Journal, JAUES* 7.2, pp. 159–172 (cit. on p. 14).

Mohtar, Hani (2010). 'Increasing surge margin of turbocharger centrifugal compressor automotive application'. PhD thesis. Nantes, France: Ecole Centrale de Nantes. 241 pp. (cit. on p. 14).

Mollenhauer, Klaus and Helmut Tschöke, eds. (2007). *Handbuch Dieselmotoren*. 3., neubearb. Aufl. VDI-Buch. Berlin: Springer. 702 pp. (cit. on p. 23).

Nakhjiri, Mehdi, Peter F. Pelz, et al. (2018). 'vATL – ein virtueller Abgasturbolader auf Basis physikalischer Modelle'. In: (cit. on p. 14).

Nakhjiri, Mehdi, Peter Pelz, et al. (Sept. 13, 2011). 'Physical Modeling of Automotive Turbocharger Compressor: Analytical Approach and Validation'. In: Commercial Vehicle Engineering Congress (cit. on p. 14).

Oh, H. W., E. S. Yoon, and M. K. Chung (1997). 'An optimum set of loss models for performance prediction of centrifugal compressors'. In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 211 (Part A), pp. 331–338 (cit. on p. 14).

Okhuahesogie, O. F. et al. (May 2014). 'A 1-D analytical code for the design and multi-objective optimisation of high-pressure compressors within two-stage turbochargers for marine applications'. In: *Conference Proceedings*. 11th International Conference on Turbochargers and Turbocharging. London, pp. 125–135 (cit. on p. 14).

Pampreen, R. C. (Apr. 18, 1982). 'Jet-Wake Analysis for Centrifugal Compressor Static Pressure Rise and Deviation Angle'. In: *Volume 1: Turbomachinery*. ASME 1982 International Gas Turbine Conference and Exhibit. London, England: ASME, V001T01A006 (cit. on p. 4).

Pelton, Robert J. (Dec. 3, 2007). 'One-Dimensional Radial Flow Turbomachinery Performance Modeling'. Master Thesis. Provo: Brigham Young University. 114 pp. (cit. on p. 14).

Pixberg, Christian Thilo (Apr. 11, 2013). 'Modellbildung zur Steigerung der Vorhersagegenauigkeit von Leistungsdaten und Stabilität axialer Verdichter'. PhD thesis. Darmstadt: Technische Universität (cit. on p. 14).

Rolls-Royce Power Systems AG (Oct. 20, 2011). *MTU Turbocharging* (cit. on p. 15).

Sakellaridis, Nikolaos F. et al. (Nov. 2015). 'Development and validation of a new turbocharger simulation methodology for marine two stroke diesel engine modelling and diagnostic applications'. In: *Energy* 91, pp. 952–966 (cit. on p. 14).

Schiffmann, J. and D. Favrat (Jan. 2010). 'Design, experimental investigation and multiobjective optimization of a small-scale radial compressor for heat pump applications'. In: *Energy* 35.1, pp. 436–450 (cit. on p. 14).

Schneider, M. et al. (Mar. 2015). 'Analytical Loss Prediction for Turbocharger Compressors'. In: 11th European Conference on Turbomachinery Fluid dynamics & Thermodynamics. European Turbomachinery Society (cit. on p. 14).

Schur, F. (2013). 'A Transient Model of a Turbofan Engine in Simulink'. In: *Conference Proceedings*. Deutscher Luft- und Raumfahrtkongress, pp. 1–9 (cit. on p. 14).

Schwarz, J.B. (2014). 'Considerations for gas stand measurement of turbocharger performance'. In: p. 12 (cit. on p. 14).

Serrano, J.R. et al. (Dec. 2008). 'A model of turbocharger radial turbines appropriate to be used in zero- and one-dimensional gas dynamics codes for internal combustion engines modelling'. In: *Energy Conversion and Management* 49.12, pp. 3729–3745 (cit. on p. 14).

Shaaban, S. et al. (2006). 'Part-load Performance Prediction of Turbocharged Engines'. In: *Conference Proceedings*. 8th International Conference on Turbochargers and Turbocharging. University of Hannover, Turbomachinery Laboratory: Woodhead Publishing, pp. 131–144 (cit. on p. 14).

Shaaban, Sameh (2004). 'Experimental investigation and extended simulation of turbocharger non-adiabatic performance'. PhD thesis. Hannover: Universität Hannover (cit. on p. 14).

Shakal, Chris (Nov. 8, 2015). *Centrifugal Compressor*. GrabCAD. Free use as per paragraph 6 of https://grabcad.com/terms. URL: https://grabcad.com/library/centrifugal-compressor-4 (visited on 01/17/2019) (cit. on p. 13).

Solaesa, Sergio Sanz (Sept. 2016). 'Analytical prediction of turbocharger compressor performance: a comparison of loss models with numerical data'. Master Thesis. Madrid: Universidad Politécnica de Madrid. 57 pp. (cit. on p. 14).

Stanitz, John D. (Jan. 1, 1952). *One-dimensional Compressible Flow in Vaneless Diffusers of Radial- and Mixed-flow Centrifugal Compressors, Including Effects of Friction, Heat Transfer and Area Change*. Technical Note NACA-TN-2610. Cleveland, OH, United

States: National Advisory Committee for Aeronautics. Lewis Flight Propulsion Lab., p. 62 (cit. on p. 19).

Stuart, C. et al. (2014). 'An analysis of stage performance in automotive turbocharger centrifugal compressors'. In: *Turbocharger Seminar 2014*. Turbocharger Seminar 2014. Tianjin, China, p. 10 (cit. on p. 14).

Taburri, M. et al. (Apr. 2012). 'A model-based methodology to predict the compressor behaviour for the simulation of turbocharged engines'. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 226.4, pp. 560–574 (cit. on p. 14).

Tacconi, Jacopo (Sept. 4, 2018). 'Investigation of a semi-closed cycle small gas turbine for high altitude UAV propulsion'. Master Thesis. Delft: Delft University of Technology. 115 pp. (cit. on p. 14).

Uchida, Hiroshi (Jan. 2006). 'Transient Performance Prediction for Turbocharging Systems Incorporating Variable-Geometry Turbochargers'. In: *R&D Review of Toyota CRDL* 41.3, pp. 22–28 (cit. on p. 14).

Wahlström, J. and L. Eriksson (July 2011). 'Modelling diesel engines with a variable-geometry turbocharger and exhaust gas recirculation by optimization of model parameters for capturing non-linear system dynamics'. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 225.7, pp. 960–986 (cit. on p. 14).

Watel, Eric et al. (Apr. 12, 2010). 'Matching and Evaluating Methods for Euro 6 and Efficient Two-stage Turbocharging Diesel Engine'. In: SAE 2010 World Congress & Exhibition (cit. on p. 14).

Whitfield, A. (May 1990). 'Preliminary Design and Performance Prediction Techniques for Centrifugal Compressors'. In: *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy* 204.2, pp. 131–144 (cit. on p. 14).

Whitfield, A. and N. C. Baines (1990). *Design of Radial Turbomachines*. Harlow, Essex, England : New York, NY: Longman Scientific & Technical ; Wiley. 397 pp. (cit. on p. 13).

Xu, Cheng and Ryoichi S. Amano (2012). 'Empirical Design Considerations for Industrial Centrifugal Compressors'. In: *International Journal of Rotating Machinery* 2012, pp. 1–15 (cit. on p. 14).

Yang, Xiaojian and Guoming G. Zhu (June 2010). 'A mixed mean-value and crank-based model of a dual-stage turbocharged SI engine for Hardware-In-the-Loop simulation'. In: *Proceedings of the 2010 American Control Conference*. 2010 American Control Conference (ACC 2010). Baltimore, MD: IEEE, pp. 3791–3796 (cit. on p. 14).

Zahn, Sebastian (2012). 'Arbeitsspielaufgelöste Modellbildung und Hardware-in-the-Loop-Simulation von Pkw-Dieselmotoren mit Abgasturboaufladung'. PhD thesis. Darmstadt: Technische Universität Darmstadt (cit. on p. 14).

Zhuge, W. et al. (May 1, 2009). 'Development of an advanced turbocharger simulation method for cycle simulation of turbocharged internal combustion engines'. In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 223.5, pp. 661–672 (cit. on p. 14).

Further Reading

These references were used for this thesis but not cited in the main body; they are provided here as-is.

Bohl, Willi and Wolfgang Elmendorf (2014). *Technische Strömungslehre*. 15., überarbeitete und erweiterte Auflage. Vogel Fachbuch. Würzburg: Vogel Buchverlag. 504 pp.

Buck, Arden L. (Dec. 1, 1981). 'New Equations for Computing Vapor Pressure and Enhancement Factor'. In: *Journal of Applied Meteorology* 20.12, pp. 1527–1532.

Davidson, Thomas A. (Jan. 1, 1993). *A simple and accurate method for calculating viscosity of gaseous mixtures*. Report of Investigations 9456. Amarillo, Texas: United States Department of the interior, Bureau of Mines.

Dehner, Rick et al. (May 10, 2016). 'Simulation of Deep Surge in a Turbocharger Compression System'. In: *Journal of Turbomachinery* 138.11, pp. 1–12.

Graham, Thomas (Jan. 1846). 'On the Motion of Gases'. In: *Philosophical Transactions of the Royal Society of London* 136 (XXVIII.), pp. 573–631.

Grieb, Hubert (2004). *Projektierung von Turboflugtriebwerken*. 1st ed. Technik der Turboflugtriebwerke. MTU München. Basel: Birkhäuser Basel. 826 pp.

– (2009). *Verdichter für Turbo-Flugtriebwerke*. 1st ed. MTU München. Berlin Heidelberg: Springer. 696 pp.

Hendricks, Elbert (Mar. 1, 2001). 'Isothermal vs. Adiabatic Mean Value SI Engine Models'. In: *IFAC Proceedings Volumes*. 3rd IFAC Workshop on Advances in Automotive Control 2001, Karlsruhe, Germany, 28-30 March 2001 34.1, pp. 363–368.

Herwig, Heinz (2016). *Strömungsmechanik: Einführung in die Physik von technischen Strömungen*. 2., überarbeitete und erweiterte Auflage. Lehrbuch. Wiesbaden: Springer Vieweg. 293 pp.

Joos, Franz (2016). *Strömungsmaschinen*. 16th ed. Antriebstechnik. Hamburg: Helmut-Schmidt-Universität Hamburg.

Mathworks (2018). *Constant volume pneumatic chamber based on ideal gas law*. Mathworks Simulink Documentation. URL: https://uk.mathworks.com/help/physmod/simscape/ref/constantvolumepneumaticchamber.html (visited on 07/22/2018).

Schiff, Johannes (June 2013). 'A Preliminary Design Tool for Radial Compressors'. Master Thesis. Lund, Sweden: Lund University. 155 pp.

Shomate, C. Howard (Apr. 1954). 'A Method for Evaluating and Correlating Thermodynamic Data'. In: *The Journal of Physical Chemistry* 58.4, pp. 368–372.

Sutherland, William (Dec. 1893). 'The viscosity of gases and molecular force'. In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 36.223, pp. 507–531.

Van den Braembussche, R. A. (2018). *Design and Analysis of Centrifugal Compressors*. Newark: John Wiley & Sons, Incorporated. 408 pp.

Vogel, Hans (1914). 'Über die Viskosität einiger Gase und ihre Temperaturabhängigkeit bei tiefen Temperaturen'. In: *Annalen der Physik* 348.8, pp. 1235–1272.

Yamaguchi, H. (2008). *Engineering fluid mechanics*. Fluid mechanics and its applications v. 85. Dordrecht, The Netherlands: Springer. 573 pp.

Indices