Optimization in Manufacturing Engineering

ÉCOLE CENTRALE DE NANTES

M2 ADVANCED MANUFACTURING

Optimization Project

Authors:
Sudarson NANTHACOUMARANE
Ahmad MOHAMAD

Date: December 23, 2021

Contents

List of Equations

1. First Section

This starts an indented paragraph. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras malesuada, lectus ac fermentum pharetra, odio mi condimentum arcu, in ultrices dolor est hendrerit massa. Vestibulum sed sagittis metus. Donec velit risus, ultrices vel pretium nec, commodo id nisi. References can be added.

This starts an unindented paragraph. Nunc lobortis mauris in semper eleifend. Sed tortor quam, pharetra vitae convallis quis, dignissim in nisl. Curabitur sed leo ac urna posuere pharetra quis eu sem. Curabitur molestie sapien ac felis ultricies varius. Nullam varius sapien eros, non imperdiet neque mattis pharetra. Nulla eleifend vel nisl sed sagittis. Donec eu dolor neque. Vivamus tempus commodo vestibulum. Mauris mattis quam ante.

Here is proof that forward referencing works. Figure 2.1.1 shows an old monument in Bharathi Park, Pondicherry.

1.1. Pythogoram Theorem

1.2. Pythogoram Theorem

1.3. Pythogoram Theorem

1.4. Pythogoram Theorem

1.5. Pythogoram Theorem

1.6. Pythogoram Theorem

1.7. Pythogoram Theorem

1.8. Pythogoram Theorem

This theorem states that given a triangle with sides a, b, and $c \Rightarrow\ a^2 + b^2 = c^2$

2. Second Section

Here, we take a look at indentation[1]

2.1. Subsection

- Bullet point 1

- Bullet point 2

$$f(x) = 5x + 3 \tag{1}$$

$$f(x) = 5x + 3 \tag{2}$$

$$f(x) = 5x + 3 \tag{3}$$

$$f(x) = 5x + 3 \tag{4}$$

$$f(x) = 5x + 3 \tag{5}$$

$$f(x) = 5x + 3 \tag{6}$$

$$f(x) = 5x + 3 \tag{7}$$

$$f(x) = 5x + 3 \tag{8}$$

$$f(x) = 5x + 3 \tag{9}$$

$$f(x) = 5x + 3 \tag{10}$$

$$f(x) = 5x + 3 \tag{11}$$

$$f(x) = 5x + 3 \tag{12}$$

$$e^{(i\theta)} = 1 + e^{(i\theta)} + \frac{1}{2!}(i\theta)^{2\cdots} \tag{13}$$

$$+ \frac{1}{N-1}(i\theta)^{N-1} + \frac{1}{N}(i\theta)^{N}$$

$$K_e = \int_{T_e} K \begin{bmatrix} (c_1^k)^2 & x_k & y_k \\ (c_1^l)^2 & x_l & y_l \\ (c_1^m)^2 & x_m & y_m \end{bmatrix} d\Omega \tag{14}$$

$$a^2 + b^2 = c^2$$

$$E = mc^2 \tag{15}$$

```
def myfunction(arg):
    arg = arg**2 - arg + 1
    return int(arg)
```

Figure 1: Bharathi Park

Figure 2: CAPTION

3. Last Section

In this section, we will look at tables and PDF links.[2]

3.1. Links using .pdfhref

3.1.1. Internet Hyperlinks

Here is an internet hyperlink to the Groff Manual where you can find documentation for groff.
Youtube is a great website for informational videos

3.1.2. PDF links

- Affixed text to Equation (2.1.1) is the first equation
- Fig(2.1.2) is the MSI logo
- Table(3.2.1) contains the specs for carburetors found in Yamaha DT 125 motorcycle.
- **Notice that this does forward referencing.**

3.2. Table example

Yamaha DT 125 Carburetor Specifications			
Bike Model	**TZR**	**DT 3DBI**	**DT 3RN1 onward**
Make	Mikuni	Mikuni	Mikuni
Type	VM26SS	VM26SS	VM26SS
ID Mark	2RH00	3BN00	3MB00
Main Jet	180	125	210
Air Jet	0.8	0.8	0.8
Jet Needle	406	407	5J25
Needle clip position	4th	3rd	4th
Float height - all models	20-21mm (0.787-0.827in)		

Table 1: Carburetor specifications for Yamaha DT 125

Figure 3: Arch

This is some code:
a
b
c

```
#!/bin/sh
# This script will compile or run another finishing operation on a docume    nt. I
# have this script run via vim.
#
# Compiles .tex. groff (.mom, .ms), .rmd, .md.  Opens .sent files as sent
# presentations.  Runs scripts based on extention or shebang
#
# Note that .tex files which you wish to compile with XeLaTeX should have     the
# string "xelatex" somewhere in a comment/command in the first 5 lines.

file=$(readlink -f "$1")
dir=${file%/*}
base="${file%.*}"
ext="${file##*.}"

clear

cd "$dir" || exit

textype() {
    command="pdflatex"
    ( sed 5q "$file" | grep -i -q 'xelatex' ) && command="xelatex"
    $command --output-directory="$dir" "$base" &&
    grep -i addbibresource "$file" >/dev/null &&
    biber --input-directory "$dir" "$base" &&
    $command --output-directory="$dir" "$base" &&
    $command --output-directory="$dir" "$base"
}

file=$(readlink -f "$1")
dir=${file%/*}
base="${file%.*}"
ext="${file##*.}"

clear

cd "$dir" || exit

textype() {
    command="pdflatex"
    ( sed 5q "$file" | grep -i -q 'xelatex' ) && command="xelatex"
    $command --output-directory="$dir" "$base" &&
    grep -i addbibresource "$file" >/dev/null &&
    biber --input-directory "$dir" "$base" &&
    $command --output-directory="$dir" "$base" &&
    $command --output-directory="$dir" "$base"
}
file=$(readlink -f "$1")
dir=${file%/*}
```

```
base="${file%.*}"
ext="${file##*.}"

clear

cd "$dir" || exit

textype() {
    command="pdflatex"
    ( sed 5q "$file" | grep -i -q 'xelatex' ) && command="xelatex"
    $command --output-directory="$dir" "$base" &&
    grep -i addbibresource "$file" >/dev/null &&
    biber --input-directory "$dir" "$base" &&
    $command --output-directory="$dir" "$base" &&
    $command --output-directory="$dir" "$base"
}
#!/bin/sh
# This script will compile or run another finishing operation on a docume    nt. I
# have this script run via vim.
#
# Compiles .tex. groff (.mom, .ms), .rmd, .md.  Opens .sent files as sent
# presentations.  Runs scripts based on extention or shebang
#
# Note that .tex files which you wish to compile with XeLaTeX should have     the
# string "xelatex" somewhere in a comment/command in the first 5 lines.

file=$(readlink -f "$1")
dir=${file%/*}
base="${file%.*}"
ext="${file##*.}"

clear

cd "$dir" || exit

textype() {
    command="pdflatex"
    ( sed 5q "$file" | grep -i -q 'xelatex' ) && command="xelatex"
    $command --output-directory="$dir" "$base" &&
    grep -i addbibresource "$file" >/dev/null &&
    biber --input-directory "$dir" "$base" &&
    $command --output-directory="$dir" "$base" &&
    $command --output-directory="$dir" "$base"
}

file=$(readlink -f "$1")
dir=${file%/*}
base="${file%.*}"
ext="${file##*.}"

clear

cd "$dir" || exit

```

```
textype() {
    command="pdflatex"
    ( sed 5q "$file" | grep -i -q 'xelatex' ) && command="xelatex"
    $command --output-directory="$dir" "$base" &&
    grep -i addbibresource "$file" >/dev/null &&
    biber --input-directory "$dir" "$base" &&
    $command --output-directory="$dir" "$base" &&
    $command --output-directory="$dir" "$base"
}
file=$(readlink -f "$1")
dir=${file%/*}
base="${file%.*}"
ext="${file##*.}"

clear

cd "$dir" || exit

textype() {
    command="pdflatex"
    ( sed 5q "$file" | grep -i -q 'xelatex' ) && command="xelatex"
    $command --output-directory="$dir" "$base" &&
    grep -i addbibresource "$file" >/dev/null &&
    biber --input-directory "$dir" "$base" &&
    $command --output-directory="$dir" "$base" &&
    $command --output-directory="$dir" "$base"
}
```

```
function docs
 This is a pseudo-function
    ...
    Function does stuff
end
```

List of Figures

List of Tables

References

1. M O Tatar and A Pop, “Development of an in pipe inspection minirobot,” *IOP Conf. Series: Materials Sceince and Engineering* **147** (2016).
2. By Atsushi Kakogawa and Shugen Ma, “Robotic Search and Resque through In-Pipe Movement” in*Unmanned Robotic Systems and Applications,* ed. Mahmut Reyhanglu and Geert De Cubber (2019).