

Politechnika Warszawska

W Y D Z I A Ł S A M O C H O D Ó W
I M A S Z Y N R O B O C Z Y C H



Praca przejściowa / dyplomowa inżynierska / magisterska

na kierunku Wpisać kierunek

w specjalności Wpisać specjalność

Wpisać tytuł pracy dyplomowej po polsku

Wpisać tytuł pracy dyplomowej po angielsku

Wpisać imię i nazwisko

Wpisać numer albumu???

Opiekun naukowy:

Dr hab. inż. Robert Zalewskim, prof. PW

Promotor:

dr inż. Bogumił Chiliński

Konsultacje:

Wpisać konsultanta

Warszawa Wpisać rok

Wpisać imię i nazwisko

OŚWIADCZENIE

Jako autor pracy dyplomowej pt.:

"Wpisać tytuł pracy dyplomowej po polsku",

którą wykonałem/am samodzielnie przestrzegając zasad ochrony własności intelektualnej zezwalam na publiczne udostępnienie pracy i wyrażam zgodę na jej udostępnienie w Bibliotece Wydziału Samochodów i Maszyn Roboczych Politechniki Warszawskiej w ramach realizacji zadań statutowych biblioteki.

Warszawa, dnia ???.

/podpis/

SŁOWA KLUCZOWE

Słowa kluczowe: Wpisać słowa kluczowe po polsku

Streszczenie

Wpisać tytuł pracy dyplomowej po polsku

Problematyką poruszoną w niniejszej pracy było opracowanie metodyki pisania prac dyplomowych i przejściowych, realizowanych na Politechnice Warszawskiej. Rezultatem podjętych przez Autorów prac jest dokument będący jednocześnie szablonem, a także propozycją metodyki planowania prac, mającą ułatwić studentom pisanie podobnych opracowań.

Na początku opracowania przedstawiono cel oraz zakres planowanych prac, a także motywację skłaniającą do podjęcia tematyki. Kolejno opisano przyjętą metodykę badawczą. W pierwszych rozdziałach opisany został proces przygotowania do tworzenia pracy dyplomowej, w którym ujęto takie aspekty jak kompozycja pracy oraz podkreślono ważność systematyczności jej opracowywania. Kolejno omówiona została poprawność językową oraz stylistyczna tekstu. Uzupełnieniem rozdziału było przedstawienie podstawowych i powszechnie używanych narzędzi wspomagających edycję tekstu oraz obliczenia inżynierskie. W ramach rozpatrywanego zagadnienia przeprowadzono analizę porównawczą dostępnego na rynku oprogramowania do edycji tekstu oraz wspomagającego prace inżynierskie, a następnie wybrano spośród nich spełniające, założone przez Autorów pracy cele i kryteria doboru. Wybranim oprogramowaniem wspomagającym pracę redakcyjną został $\text{L}^{\text{A}}\text{T}_{\text{E}}\text{X}$, a środowisko wspomagające obliczenia inżynierskie - SageMath.

Następnie dokonano obszernego opisu składania tekstu w środowisku $\text{T}_{\text{E}}\text{X}$ oraz działania przydatnych funkcji oraz pakietów służących formatowaniu i edycji tekstu, a także tworzeniu wzorów, tabel i rysunków. Środowisko SageMath wspomagające obliczenia inżynierskie zostało wyczerpująco opracowane w kolejnym rozdziale. Omówiono w nim język skryptowy Python oraz wiele przydatnych bibliotek ułatwiających i optymalizujących prace związane z tworzeniem skryptów oraz opracowywaniem wyników. Całość pracy została podsumowana syntetycznymi wnioskami.

Słowa kluczowe: Wpisać słowa kluczowe po polsku

Abstract

Wpisać tytuł pracy dyplomowej po angielsku

At the beginning of the paper present the construction of crankshafts and identify the phenomena that must be taken into account by designers at the stage of their design. These include vibrations that can lead the shaft to resonance. The methods of suppressing this phenomenon as well as tools for vibration analysis were introduced.

The methods of deriving equations of motion were presented and the mathematical model of the position was adopted in subsequent chapters. From the model the moment of inertia and torsional stiffness was calculated. Natural frequency was initially around 5 Hz .

Then, the selection of the elements of the station is described, i. e. the selection of discs with a specific moment of inertia and the selection of springs with a specific torsional stiffness. As part of the verification of the calculations, the parameters of the elements were experimentally checked and the stand was additionally created in two variants of construction. Experiments have confirmed the formulae used in the calculations.

In the fifth chapter, a series of measurements of the lateral accelerations of the individual discs were carried out. Accelerations were measured with the free oscillating system and with the free oscillating system under the influence of extortion.

On the basis of the results of a series of measurements it can be concluded that the assumed model is good and the obtained results are qualitative and consistent with the calculations.

Keywords: Wpisać słowa kluczowe po angielsku

Spis treści

1. Wstęp	11
1.1. Wprowadzenie do tematyki	11
Czym jest praca dyplomowa	11
Istotność wstępu pracy	11
Cel pisanie pracy dyplomowej	11
1.2. Cel i zakres pracy	12
główny cel pracy - ogólny zarys	12
Dodatkowe cele pracy	12
Określenie głównych założeń	12
Zakres opracowania	12
1.3. Motywacja i aktualny stan wiedzy	13
Motywacja	13
Brak publikacji na ten temat	13
Wymogi i kryteria tworzenia pracy	13
Publikacja o pokrewnej tematyce	13
Inna publikacja o pokrewnej tematyce	13
1.4. Metodyka badawcza	13
Przedstawienie koncepcji analizy i badania	13
Analiza porównawcza dostępnych pakietów - wybór	14
Opis wybranego edytora tekstu	14
1.5. Przygotowanie do pisanie pracy dyplomowej	14
Czemu ma służyć praca dyplomowa?	14
Przygotowanie spisu treści	14
Systematyczność w przygotowywaniu pracy i egzekwowanie postępów	14
Struktura pracy dyplomowej	14

	Spójność i logika w podziale pracy	15
1.6.	Opracowanie językowe tekstu	15
	Poprawność składniowa	15
	Neutralność opisu i oceny, błędy stylistyczne	16
1.7.	Narzędzia edycji tekstu	16
	Opis wybranych programów do edytowania tekstów	16
	Alternatywy	16
1.8.	Programy wspomagające obliczenia inżynierskie	17
	Opis wybranych programów, pakietów obliczeniowych	17
2.	Oprogramowanie \LaTeX	19
	Opis oprogramowania, jego zalety, porównanie z alternatywnymi narzędziami	19
2.1.	Składania tekstu z wykorzystaniem środowiska \TeX	19
	Opis konstrukcji pliku tekstowego	19
2.2.	Struktura dokumentu	19
	Opis kodu definiującego format tekstu oraz możliwości wykonywania zadań	20
2.3.	Wykorzystywanie pakietów oraz formatowanie	20
	Opis procesu tworzenia i numerowania wzorów	20
2.4.	Wstawianie wzorów, tabel oraz rysunków	21
	Opis budowy struktury i systematyzowania rysunków	21
	Opis budowy struktury tabel	22
	Systematyzacja spisów rysunków, tabel	23
2.5.	Automatyczne generowanie spisów rzeczy	23
	Opis tworzenia spisu bibliograficznego i cytowania w tekście	23
2.6.	Tworzenie bibliografii - środowisko Bib \LaTeX	23
	Dodatkowe informacje o przydatnych funkcjach	24
2.7.	Przydatne funkcje	24
	Przykłady	24
3.	Wspomagane komputerowo obliczenia w środowisku SageMath	26
3.1.	Środowisko SageMath	26
	Czym jest SageMath	26

	Informacje podstawowe	26
	Zakres zastosowań	26
	Zalety środowiska SageMath	26
3.2.	Język skryptowy Python	27
	Co to Python?	27
	Składnia języka Python	27
	Paradygmaty i działanie	27
	język skryptowy i enkapsulacja	27
	przeznaczenia Pythona oraz obliczenia inżynierskie	27
3.3.	Biblioteka NumPy	27
	co to jest NumPy	27
	z czego się składa NumPy	27
	co umożliwi w kontekście pisania pracy dyplomowej	28
	przykład zastosowania	28
3.4.	Biblioteka Matplotlib	29
	Czym jest Matplotlib	29
	Możliwości pakietu	29
	Przygotowanie pola wykresu	29
	Przykład zastosowania	29
3.5.	Pakiet SageTeX	30
	Czy jest pakiet SageTeX	30
	Możliwości pakietu	30
	Importowanie wzorów z SageMath	31
	Zalety stosowania SageTeX	32
3.6.	Eksportowanie wyników do dokumentu \LaTeX	32
	co można i warto eksportować	32
	umieszczanie elementów w dokumencie \LaTeX	32
	Przykład osadzania wzoru w tekście	32
	Przykład osadzania grafiki	33
	Działanie finalnego dokumentu - podsumowanie rozdziału	33
4.	Podsumowanie	35

Bibliografia	36
Wykaz symboli i skrótów	37
Słownik terminów	38
Spis rysunków	40
Spis tabel	41
Spis załączników	42
Załączniki	43

1. Wstęp

1.1. Wprowadzenie do tematyki

Praca dyplomowa jest podsumowaniem wiedzy i umiejętności, jakie student powinien był nabyć w trakcie nauki na wyższej uczelni. Praca dyplomowa powinna być zatem opracowana samodzielnie i twórczo, jak również dotyczyć praktycznego lub teoretycznego problemu. Stanowi ona bowiem wizytówkę studenta, a także promotora, pod którego opieką powstawała. Zarówno student jak i promotor mają największy wpływ na jej tematykę, a także treść i formę. Należy więc zadbać o to, aby praca dyplomowa została przygotowana starannie i rzetelnie, zarówno na poziomie merytorycznym jak i redakcyjnym.

Wstęp jest jedną z najważniejszych części każdej pracy dyplomowej. Jako że jest najczęściej czytany, powinien zawierać więc krótkie omówienie problemu, będącego przedmiotem pracy oraz krótkie przedstawienie aktualnego stanu wiedzy - źródła literaturowe w zakresie poruszanej problematyki, oraz ewentualne hipotezy, które autor zamierza sprawdzić lub udowodnić. Dodatkowo należy uwzględnić motywacje, które skłoniły autora pracy do zajęcia się daną tematyką. Wszystkie te informacje mają za zadanie uzasadnić celowość podjęcia pracy oraz zaciekać i zachęcić Czytelnika do dalszego zapoznania się z tematem.

Dla wielu studentów praca dyplomowa, poza próbą wykazania się umiejętnościami formułowania i rozwiązywania problemów, wykazaniem się znajomością metod analitycznych i sposobów badawczych, stanowi pierwszą, większą wypowiedź pisemną. Służy ona zatem nauce poprawnego pisania w stylu naukowym, a także nauce poprawnego redagowania i formatowania tekstu, tak aby struktura, konstrukcja, język i stylistyka pracy pozwalały Czytelnikowi na łatwe zrozumienie omawianej przez autora tematyki. W niniejszej pracy omówione zostaną zatem środowiska wspomagające pisanie pracy dyplomowej od strony redakcyjnej i inżynierskiej. W tym celu przedstawione i opisane zostanie oprogramowanie \LaTeX , wspomagające tworzenie pracy od strony redakcyjnej oraz formatowania oraz środowisko SageMath, wraz z przydatnymi bibliotekami, jako narzędzie wspomagające obliczenia inżynierskie.

1.2. Cel i zakres pracy

Celem pracy jest opracowanie metodyki wykonywania prac i pisania tekstów inżynierskich jakimi są prace dyplomowe i przejściowe realizowane na Politechnice Warszawskiej. Proces przygotowania takiego opracowania składa się z dwóch aspektów, które najczęściej wykonywane są równolegle, rzadziej sekwencyjnie. Pierwszym aspektem jest realizacja celu, stanowiąca wartość dodaną. Drugi element to opis przeprowadzonych działań, który to w większości jest wymiernym efektem prowadzonych prac. W związku z tym, jakość tej części w sposób istotny wpływa na ocenę całego dzieła.

Ponadto planowanym rezultatem przeprowadzonej pracy jest gotowy dokument będący jednocześnie szablonem tezy oraz pomocą mającą ułatwić studentom pisanie podobnych opracowań. W praktyce okazuje się, że studenci potrzebują dodatkowych materiałów pomocniczych. Przyczyną jest konieczność spełnienia wielu wymagań jakie stawia się tego typu tekstom oraz brak jednoznacznych wytycznych odnośnie do wszystkich aspektów występujących w pracach realizowanych na Politechnice Warszawskiej.

Praktyka wskazuje, że większość opracowań o charakterze inżynierskim lub naukowym wykazuje pewne podobieństwo. Bazując tylko na wiedzy i doświadczeniu Autorów przyjęto, że analizowane prace charakteryzują się:

- stałą kompozycją,
- występowaniem wzorów i zależności matematycznych,
- koniecznością wykonywania analiz oraz obliczeń,
- obszernym materiałem ilustracyjnym w postaci zdjęć i wykresów,
- formą typową dla tekstów naukowych.

W związku z tym, założono że w ramach tej pracy przygotowany zostanie tekst:

- zgodny z wymaganiami stawianymi opracowaniom inżynierskim lub naukowym realizowanym na Politechnice Warszawskiej,
- zawierający wszystkie elementy typowe dla tego typu dzieł,
- automatyzujący często powtarzające się działania.

Wymiernym efektem przeprowadzonych prac jest dokument będący jednocześnie propozycją metodyki opracowywania tez oraz edytowalnym szablonem pracy dyplomowej.

1.3. Motywacja i aktualny stan wiedzy

Motywacją do stworzenia niniejszego szablonu była chęć pomocy studentom przy samodzielnym opracowywaniu pracy dyplomowej. Ze względu na niezadawalający poziom stylistyczny, a także kłopoty przy redagowaniu i formatowaniu prac tworzonych dotychczas przez studentów, autorzy niniejszego szablonu zdecydowali się na przygotowanie krótkiego opracowania mającego na celu ułatwienie studentom samodzielne redagowanie i edytowanie tekstów prac dyplomowych czy przejściowych. Jednocześnie jednak, szablon ma stanowić pomoc promotorom, ograniczając ich pracę wyłącznie do korekty merytorycznej, a nie jak miało to miejsce do tej pory, redakcyjnej, która może zostać wykonana samodzielnie przez studenta, już na etapie tworzenia pracy.

Jako, że do tej pory na Wydziale Samochodów i Maszyn Roboczych, poza oficjalnymi wytycznymi pisania prac dyplomowych zamieszczonymi w zarządzeniu Rektora PW oraz informacjami, jakie można znaleźć na stronie Wydziału SiMR, nie powstała jeszcze praca o podobnej tematyce, autorzy tekstu pokładają ufność, iż dzięki ich wysiłkom, od teraz tworzenie, redagowanie i formatowanie tekstów prac dyplomowych będzie dla studentów dużo bardziej przystępne, a staranność nowopowstałych prac na najwyższym poziomie.

Prace dyplomowe oraz przejściowe charakteryzują się pewnymi cechami związanymi ze stylem, językiem czy edycją tekstu. Właścicielem pracy wykonanej na uczelni, w ramach pracy dyplomowej jest jednostka naukowa. W przypadku prac tworzonych na wydziale Samochodów i Maszyn Roboczych, to właśnie wydział jest ich właścicielem. Z tego powodu narzucone są określone wymagania związane z wyglądem pracy. Wszelkie informacje dotyczące edycji tekstu, zawartości pracy czy innych wymogów są do znalezienia na stronie internetowej Wydziału SiMR: [ODNOŚNIK]. Oficjalne, ujednolicone wytyczne zawierają zarządzenia Rektora Politechniki Warszawskiej nr 43/2016 z dnia 8/09/2016 i nr 57/2016 z dnia 15/12/2016 [odnośnik][odnośnik]

1.4. Metodyka badawcza

Planuje się w ramach rozwiązywania rozpatrywanego zagadnienia przeprowadzenie analizy porównawczej dostępnego na rynku oprogramowania do wspomagania prac inżynierskich.

następnie wybranie tych, które spełniać będą założone przez autorów pracy cele i kryteria doboru. K a także dwóch wybranych spośród wielu, pakietów optymalizujących pracę studenta. Jednym z nich będzie oprogramowanie wspomagające pracę redakcyjną oraz środowisko wspomagające obliczenia inżynierskie. Pozwoli to na zaproponowanie łatwego procesu generowania dokumentów spełniających

stawiane pracom dyplomowym warunki.

1.5. Przygotowanie do pisania pracy dyplomowej

Przed rozpoczęciem pisania pracy dyplomowej należy pracę tę zaplanować, przemyśleć jej zakres oraz poznać wytyczne dotyczące zawartości i formatowania tekstu. Sama praca dyplomowa ma na celu udowodnienie posiadania przez studenta umiejętności samodzielnego rozwiązywania problemu inżynierskiego lub naukowego, w oparciu o zdobytą w trakcie studiów wiedzę. Tematyka prac powinna być związana problemami technicznymi dotyczącymi budowy i eksploatacji maszyn, pojazdów itp. Dobrze, kiedy praca dyplomowa traktuje o nietypowych zagadnieniach czy rozwiązuje rzeczywisty problem.

W tym miejscu należy zaznaczyć, że właścicielem praw do pracy dyplomowej jest Wydział Samochodów i Maszyn Roboczych Politechniki Warszawskiej.

Dobrym przygotowaniem i planem pracy jest spis treści. Dokładne przemyślenie spisu treści pozwala zapanować nad logiką pracy, wyznacza kierunki jej rozwoju, zbiera informacje dotyczące koniecznej do zastosowania teorii. W pierwszych fazach powstawania pracy, sugeruje się aby spis treści był podzielony na jak najwięcej konkretnych punktów. Każdy punkt powinien wskazywać na jedno, wyspecjalizowane zagadnienie. Przykład: jeżeli rozdział o oscylatorze harmonicznym ma zawierać rysunek, opis, równania i podsumowanie, należy stworzyć cztery punkty używając środowiska \LaTeX warto posłużyć się komendą `\aspect`. Prowadzi to do podziału pracy na rozdziały (`\chapter`), podrozdziały (`\section`) i niewyszczególnione w docelowym spisie treści punkty - zagadnienia (`\aspect`).

Ułożenie przemyślanego i skonsultowanego z opiekunem spisu treści ułatwia zachowanie systematyczności w wykonywaniu prac. Postępy w tworzeniu pracy są egzekwowane i oceniane przez opiekuna. Niewywiązywanie się z ustalonych terminów i jawne ignorowanie zasad pisania prac dyplomowych skutkuje brakiem możliwości ponadnormatywnego konsultowania pracy, a nawet wydaleniem z grona dyplomantów będących pod opieką danego opiekuna. Ponadto, spis treści pozwala przezwyciężyć trudności związane z utrzymaniem uwagi na temacie pracy.

Struktura pracy dyplomowej wykonanej na Wydziale Samochodów i Maszyn Roboczych jest określona przez wymieniony w rozdziale pierwszym dokument. Dyplom ma zawierać następujące elementy (podane w kolejności występowania):

1. Strona tytułowa - zgodna z szablonem strony tytułowej dostępnym na stronie APD,
2. Oświadczenie autora,

3. Słowa kluczowe - 3-4 słowa pozwalające łatwo odnaleźć pracę i rozwiązywany w niej problem,
4. Streszczenie w języku polski i w języku obcym - obie wersje muszą być jednakowe co do treści. Pod streszczeniami musi znaleźć się odręczny podpis autora i prowadzącego pracę,
5. Spis treści - w przypadku pracy zespołowej należy jednoznacznie zaznaczyć wykonawców poszczególnych rozdziałów,
6. Kolejne rozdziały pracy
7. Bibliografia,
8. Wykaz symboli i skrótów,
9. Spis rysunków,
10. Spis tabel,
11. Spis załączników,
12. Załączniki

Szczegółowy opis poszczególnych rozdziałów znajduje się w innych częściach niniejszego opracowania.

Praca powinna być podzielona w sposób logiczny, pozwalający na bezproblemowe zapoznanie się z tokiem rozumowania autora. Konieczne jest jasne sformułowanie celu i motywacji pracy. Należy zawrzeć opisy niezbędnych narzędzi i kroków wykonanych w celu rozwiązania postawionego problemu. Całość powinna być podsumowana nietrywialnymi wnioskami, które można wysnuć na podstawie przeprowadzonych działań.

1.6. Opracowanie językowe tekstu

Każde opracowanie w formie pisemnej należy tworzyć zachowując jeden czas, osobę i formę zdań. Zalecanymi do użycia środkami językowymi są:

- Czas przeszły, o autorze wypowiada się w trzeciej osobie. Przykład: *Autor wykorzystał istniejącą infrastrukturę badawczą. Dzięki niej przeprowadzono szereg eksperymentów.*
- Używanie biernego trybu czasowników. Przykład: *Eksperyment został przeprowadzony.*

Należy powstrzymać się od osobistych ocen faktów opisywanych w pracy, niepotrzebnego skalowania przymiotników oraz wyrazów uważanych za kolokwializmy i słownictwa tzw. warsztatowego. Autor nie powinien za wszelką cenę unikać powtórzeń. Praca inżynierska czy magisterska z zakresu nauk ścisłych wymaga precyzyjnego wypowiedzania się. Często próba zastąpienia ścisłego terminu fizycznego innym, prowadzi do nieporozumienia albo wręcz do zmiany sensu zdania.

1.7. Narzędzia edycji tekstu

Najpopularniejszym narzędziem do tworzenia prac dyplomowych jest środowisko "MS Office" i zawarty w nim edytor tekstu "Word". Jest to program działający w formule "What you see is what you get" (WYSIWYG), co znaczy "to co widzisz jest tym co otrzymasz". Na tej samej zasadzie działają inne edytory tekstu zawarte w pakietach takich jak OpenOffice, LibreOffice, Google Docs czy Office Online (dostępny dla studentów Politechniki Warszawskiej). Mimo wielu zalet i wygody polegającej na możliwości ciągłego podglądu pracy, tego typu edytory mają swoje wady. Korzystanie z tego oprogramowania stwarza kłopoty podczas edycji długiego tekstu, podziału stron, umieszczania rysunków, numeracji wzorów czy ilustracji.

Alternatywą dla systemów "WYSIWYG" jest środowisko \LaTeX i jemu podobne. Jest ono szczególnie użyteczne w przypadkach prac zawierających wiele wzorów matematycznych, rysunków, podziałów logicznych na akapity i podakapity. \LaTeX pozwala na automatyzację wielu operacji takich jak numeracja wzorów, cytowania pozycji literaturowych. W kombinacji z platformami integrującymi środowiska obliczeniowe z edytorami tekstu takie jak CoCalc, otrzymuje się narzędzie pozwalające na daleko posuniętą automatyzację pracy. Możliwe jest importowanie do tekstu wyników symulacji, rozwiązanych równań czy wieloelementowych macieży. Znacząco skraca to czas potrzebny na wprowadzenie takich obiektów do opracowania. Więcej na temat \LaTeX znajduje się w dalszych rozdziałach niniejszego opracowania.

Porównując do siebie środowiska do edycji tekstów WYSIWYG i środowiska typu \LaTeX do opracowywania prac dyplomowych wybrano te drugie, jako usprawniające przebieg prac. Dodatkowo, środowisko CoCalc pozwala na równoległą edycję dowolnych plików przez kilku użytkowników. Stwarza to nowe pola do współpracy między studentem a opiekunem naukowym.

1.8. Programy wspomagające obliczenia inżynierskie

W trakcie prac naukowych czy technicznych często pojawiają się problemy związane z rozwiązywaniem złożonych zagadnień. Z pomocą przychodzą programy wspomagające obliczenia inżynierskie. Problemy, które dzięki nim można rozwiązać są bardzo zróżnicowane. Oprogramowanie tego typu jest nieocenioną pomocą przy skomplikowanych obliczeniach symbolicznych, numerycznych czy przy analizie dużych zbiorów danych. Dalsza część podrozdziału służy krótkiemu opisowi przykładowych programów.

1. **MatLab** jest popularnym oprogramowaniem służącym do obliczeń numerycznych. Konsekwencją swojej powszechności jest duże rozbudowanie środowiska, obecność wielu wyspecjalizowanych narzędzi i makr do obliczeń konkretnych problemów (np: analiza statystyczna), mnogość poradników, podręczników, filmów instruktażowych itp. Cechą charakteryzującą MatLaba jest traktowanie każdej zmiennej jako macierzy. Ma to wpływ na logikę tworzenia kodów. Pakiet MatLab zawiera program Simulink, który pozwala w graficzny sposób modelować problemy inżynierskie takie jak rozwiązywanie równań różniczkowych. Dla studentów Politechniki Warszawskiej pakiet MatLab jest dostępny za darmo.
2. **Octave, SciLab** są to programy bliźniaczo podobne do MatLaba. Jedyne SciLab korzysta z nieco odmiennej składni. Logika jednak pozostaje ta sama. Oba produkty są programami typu *open source*.
3. **Python** jest językiem programowania wysokiego poziomu. Nadaje się do wielu zastosowań. Założeniem jego twórców jest łatwość i klarowność pisanego kodu. Python jest standardowo wyposażony w wiele użytecznych bibliotek. **SageMath** jest stworzony w języku Python i służy do wspomagania operacji matematycznych. Więcej informacji na temat Pythona i SageMath, jako jednych z użyteczniejszych środowisk, zostanie przedstawione w dalszej części pracy.
4. **MathCad** jest programem służącym do obliczeń algebraicznych. Jego zaletą jest przejrzystość interfejsu i łatwość przygotowywania obliczeń. Program pozwala na przeprowadzanie obliczeń symbolicznych i numerycznych.
5. **Mathematica** to komercyjny system obliczeń symbolicznych i numerycznych opracowany w 1988 przez Stephena Wolframa. W ciągu swojego istnienia stał się popularny w środowisku naukowców i inżynierów. Mathematicę charakteryzują wysoka wydajność, szerokie możliwości wizualizacji i prezentacji danych oraz przenośność. Obecnie jest dostępna na większość platform

32- i 64-bitowych. Komercyjną konkurencją dla pakietu Mathematica jest Maple. Spośród wolnego oprogramowania podobne funkcje oferują programy Maxima oraz Sage.

6. **Maxima** jest programem do obliczeń symbolicznych. W swoim działaniu jest podobna do programu Mathematica. Zaletą Maximy jest to, że jest to program darmowy.
7. **Ansys, Abaqus** są to programy do tworzenia analiz metodą elementów skończonych. Pozwalają na tworzenie modeli, importowanie geometrii z programów CAD, dyskretyzację (tworzenie siatki), ustalanie warunków brzegowych, określanie scenariusza symulacji (np: obciążenia). Z tych programów możliwe jest uzyskanie między innymi takich informacji jak naprężenia, odkształcenia, częstości własne, postacie drgań.
8. **Comsol** jest programem bazującym na metodzie elementów skończonych. Jest podobny do programów takich jak Ansys czy Abaqus. Różni się od nich tym, że w łatwy sposób umożliwia przeprowadzenie analiz układów w których występują zjawiska z różnych dziedzin fizyki np: sprzężenia elekto-chemiczne, termo-mechaniczne. Zaletami Comsola są również łatwy dostęp do równań wykorzystywanych przy obliczeniach oraz możliwość wpływania na dowolny parametr układu. Jest to oprogramowanie skierowane *stricte* dla naukowców, przez co jest mało popularne i trudno dostępne.

2. Oprogramowanie \LaTeX

2.1. Składania tekstu z wykorzystaniem środowiska \TeX

Obecnie każdy użytkownik komputera jest zmuszony do częstego pisania dokumentów tekstowych w postaci sprawozdań, wniosków, podań, prac dyplomowych itd. W tym celu powstało wiele programów komputerowych, które w różny sposób pomagają rozwiązywać problemy edytorskie. Najbardziej znanym program jest Microsoft Word. Już od najmłodszych lat w szkołach uczy się tego oprogramowania. Jednak w trakcie wykonywania złożonych problemów takich jak obliczenia inżynierskie oraz pisanie prac dyplomowych napotkano wiele problemów, które wymagają innego podejścia niż oferuje oprogramowanie. W zawodowym środowisku edytorskim (np. składanie książek oraz artykułów naukowych) powszechnie stosowanym oprogramowaniem jest system \LaTeX . Jest to forma pisania tekstu poprzez programowanie. Dla początkujących osób może się ona wydawać nieprzyjaznym środowiskiem. Jednakże po zdobyciu podstawowej wiedzy na temat tego programu można zaobserwować potencjał oferowany przez środowisko \LaTeX . W ramach pisania pracy dyplomowej zdecydowano się na korzystanie z platformy Cocalc. Jest to środowisko sprzęgające ze sobą możliwość pisania dokumentów w \LaTeX oraz wykonywania złożonych obliczeń inżynierskich w środowisku SageMath lub z wykorzystaniem języka Python.

2.2. Struktura dokumentu

Oprogramowanie \LaTeX umożliwia tworzenie dokumentów tekstowych. Niestety nie przypomina pod względem graficznym dobrze znanego większości osób programu 'Word'. Strukturę pisania dokumentu można podzielić na kilka etapów:

- Ogólne określenie formatu dokumentu (ustawienie czcionki, rozmiaru, marginesów, podział na kolumny itd.)
- wczytanie odpowiednich paczek oraz pakietów umożliwiających wykonanie wszystkich dalszych czynności pisania pracy (np. pisanie wzorów, wstawianie rysunków, cytowanie bibliografii itd.)

- zamieszczanie tekstu, wzorów, rysunków i innych elementów według schematów opisanych w kolejnych rozdziałach,
- tworzenie bibliografii, spisu tabel, rysunków oraz załączników,
- możliwość ciągłego podglądania dokumentu w postaci pliku .pdf

Forma dokumentu została przygotowana według wytycznych przedstawionych przez komisję dyplomową. Zadaniem każdego studenta jest wypełnienie dokumentu tekstem według schematów przedstawianych w tej pracy.

2.3. Wykorzystywanie pakietów oraz formatowanie

Formuła umożliwiająca wykorzystanie odpowiedniego pakietu składa się z `\usepackage{nazwa odpowiedniej paczki}`. Na początku nauki oprogramowania \LaTeX problem może stanowić brak wiedzy w kwestii wczytywania odpowiednich paczek. Dlatego też w tym celu należy skorzystać z literatury ([1])

Formatowanie dokumentu można wykonać w sposób globalny oraz lokalny. Pierwszą komendą dokumentu .tex jest `\documentclass{klasa}`.

W dalszej części pracy istnieje możliwość zmiany formatu poszczególnych tekstów m.in:

- wielkość czcionki,
- styl,
- pogrubienie,
- kursywa,
- wyśrodkowanie,
- inne.

Składnie funkcji pozwalające zmienić lokalnie wygląd tekstu zamieszczono w rozdziale - przydatne funkcje.

2.4. Wstawianie wzorów, tabel oraz rysunków

Tworząc pracę dyplomową niezbędne jest przedstawienie wzorów pozwalających rozwiązać określony problem inżynierski. Aby wstawić wzór matematyczny należy odpowiednią komendą `\begin{equation}` rozpocząć oraz `\end{equation}` zakończyć strukturę wzoru. Następnie wykorzystując odpowiednią składnię należy zbudować docelowy wzór. W tym celu pomocne będzie korzystanie ze stron internetowych [2] i [3]. Do każdego wzoru należy dodać etykietę wykorzystując funkcję `\label{eq:nazwaetykiety}`. Nazwa etykiety powinna być unikatowa, ponieważ wyeliminuje to powtórzenia w trakcie wstawiania odnośników do tekstu. Odwołanie do wzoru wykonuje się poprzez wstawienie w tekście komendy `\ref{eq:nazwaetykiety}`. Przestrzeganie powyższych zasad powoduje automatyczne numerowanie wzoru, a dodawanie unikatowych odnośników wyeliminuje niebezpieczeństwo popełnienia błędu w cytowaniu. Jeżeli we wzorze wystąpi nowy symbol należy go opisać. W tym celu należy użyć funkcji `\begin{describe}` i `\end{describe}`.

$$n = \frac{E}{c\varphi} = \frac{U - R_t \cdot I_t}{c\varphi} \quad (2.1)$$

gdzie:

x^2 współrzędna

π - ratio of circumference of circle to its diameter

Podobnie wygląda wstawianie do dokumentu rysunków. Każdy rysunek powinien znaleźć się w tym samym folderze, co plik dokumentu \LaTeX . Składania wstawiania rysunków jest analogiczna jak w przypadku wzorów. Również należy rozpocząć komendą `begin`. Litera H w nawiasie kwadratowym powoduje wstawienie rysunku w miejsce gdzie wpisana jest struktura. Jest to istotne ze względu na to, że \LaTeX ma tendencje do wykorzystywania wolnej przestrzeni dokumentu, a wpisanie np. małej litery h spowoduje ustawienie wzoru w optymalne miejsce na stronie, niekoniecznie w miejscu, w którym znajduje się struktura. Funkcja `\centering` powoduje wyśrodkowanie rysunku oraz jego podpisu. Wstawianie odpowiedniego rysunku wywołuje się poprzez komendę `\includegraphics`. W nawiasie klamrowym należy podać źródło pliku. Rysunki należy zapisywać w rozszerzeniu .jpg. Funkcja `caption` pozwala na dodanie podpisu pod rysunkiem. Komenda `Label` tak jak w przypadku wzorów umożliwia nadanie unikatowej etykiety. Cytowanie rysunku odbywa się poprzez wstawienie komendy `\ref{fig:nazwa etykiety}`. Na koniec należy całość zakończyć przez `\end{figure}`.

```
\begin{figure}[H]
```

```
\centering
```



```

\includegraphics{./media/stanowisko_real.jpg}
\caption{Gotowe stanowisko}
\label{fig:Gotowe_stanowisko}
\end{figure}

```



Rysunek 2.1: Gotowe stanowisko

Początek i koniec budowy tabeli 2.1 jest analogiczny do tworzenia wzorów czy rysunków. Nowością jest definiowanie liczby kolumn poprzez dodawanie kolejnych pozycji w składni: `| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |`. `\hline` rysuje w tabeli linię poziomą. Następnie należy wypełniać pola tabeli. Przechodzenie do następnej kolumny wykonuje się poprzez wstawienie `&`. Zakończenie wiersza i przejście do następnego umożliwia komenda `\\`. Poniżej przedstawiono pełną budowę tabeli.

```

\begin{table}[H]
\centering
\begin{tabular}{| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |}
\hline
Nazwa 1 & Nazwa 2 & Nazwa 3 & Nazwa 4 & Nazwa 5 & Nazwa 6 &
Nazwa 7 \\ \hline
0.94 & 0.97 & 0.91 & & 0.92 & & 0.95 & 0.938 & 1
\\ \hline

```

```

\end{tabular}
\caption{Wyniki eksperymentu} \label{tab:Res_exp}
\end{table}

```

Nazwa 1	Nazwa 2	Nazwa 3	Nazwa 4	Nazwa 5	Nazwa 6	Nazwa 7
0.94	0.97	0.91	0.92	0.95	0.938	1

Tablica 2.1: Wyniki eksperymentu

2.5. Automatyczne generowanie spisów rzeczy

Jednym z wymagań prac dyplomowych jest przedstawienie m.in. spisu rysunków i tabel. Operacje te można wywołać poprzez wstawienie funkcji `\listoffigures` i `\listoftables`. Należy jednak pamiętać, aby struktura wszystkich tabel oraz rysunków posiadała formę przedstawioną w pracy. W niniejszej pracy w docelowym miejscu wstawiono powyższe funkcje, a spisy wykonują się automatycznie. Stanowi to duże uproszczenie w pisaniu pracy dyplomowej, ponieważ należy pamiętać o wstawianiu poszczególnych elementów w formie przedstawionej niniejszym dokumencie.

2.6. Tworzenie bibliografii - środowisko Bib_{La}T_EX

Jednym z elementów prac dyplomowych jest bibliografia. Środowisko Latex umożliwia bezbłędne wstawianie pozycji literaturowych. W tym celu wykonano kilka kroków niezbędnych do stworzenia bibliografii:

- stworzono plik z rozszerzeniem `.bib` (na potrzeby prac dyplomowych w folderze znajduje się plik o nazwie `Elementy_bibliografia.bib` i jego należy modyfikować) oraz dodano do niego pozycje literaturowe.
- każda pozycja w pliku `.bib` rozpoczyna się od komendy `@article(indywidualna nazwa pozycji,...)` (dla artykułów) oraz `@book(indywidualna nazwa pozycji,...)` (dla książek)
- przykładowa pozycja literaturowa w pliku `.bib`: `@article{rubber_bands,title = Do rubber bands acts like springs?, url = https://www.wired.com/2012/08/do-rubber-bands-act-like-springs/, abstract = Do rubber bands acts like springs?, urldate = 26.08.08, author = Rhett`

Allain, year = 2012—

- cytowanie w tekście kolejnych pozycji z plik .bib powoduje automatyczne wstawianie odnośnika oraz pojawienie się zacytowanej pozycji w rozdziale Bibliografia.
- w tekście źródłowym w miejscu wyświetlania bibliografii wstawiono funkcję `\printbibliography`
- cytowanie pozycji odbywa się poprzez wstawienie do tekstu funkcji `\cite{indywidualna nazwa pozycji w pliku .bib}`
przykład: W artykule (`\cite{rubber_bands}`) [4] przedstawiono możliwości zastosowania...
Należy zwrócić uwagę, że taki zabieg dodaje na koniec pracy (w bibliografii) informacje o zacytowanej pozycji literaturowej.

2.7. Przydatne funkcje

W tym rozdziale przedstawiono zbiór funkcji, które pomogą sformatować tekst, wzory, tabele, rysunki i inne rzeczy w oczekiwany przez projektanta sposób. Ze względu na błędy sposobu wyświetlania specjalnych znaków w podglądzie oraz dokumencie pdf należy zwrócić uwagę na formę składni znajdującą się w pliku źródłowym (.tex)

- Wstawianie komentarzy do dokumentu źródłowego odbywa się poprzez wstawienie `%` na początku zdania,
- `\textbf{tekst}` – pogrubienie tekstu,
- `\newpage` – wstawienie nowej strony,
- I_t - wstawienie symbolu,
- wstawianie macierzy: struktura oraz jej reprezentacja graficzna:

```
$$$K={\left[\begin{array}{ccc} 8.6 & -19.7 & 14.8 \\ -19.7 & 54.4 & -56.8 \\ 14.8 & -56.8 & 98.9 \end{array}\right]} \cdot 10^4 $$$
```

$$K = \begin{bmatrix} 8.6 & -19.7 & 14.8 \\ -19.7 & 54.4 & -56.8 \\ 14.8 & -56.8 & 98.9 \end{bmatrix} \cdot 10^4$$

- `\eqref{nazwa etykiety wzoru}` – odnośnik do wzoru wyświetlany w nawiasie okrągłym ,
- `\ref{nazwa etykiety wzoru}` – odnośnik do wzoru wyświetlany bez nawiasu,
- Aby użyć innych funkcji należy skorzystać z książek [5] i [1] oraz stron internetowych [2] i [3]
- Przykładowe wzory zamieszczono w załącznikach (na końcu pracy).

3. Wspomagane komputerowo obliczenia w środowisku SageMath

3.1. Środowisko SageMath

Środowisko SageMath (wcześniej SAGE od ang. Software for Algebra and Geometry Experimentation) jest programem typu CAS (System algebry komputerowej z ang. Computer algebra system). Autorem jest William A. Stein, który zaproponował koncepcję integracji wielu specjalistycznych narzędzi, zamiast tworzenia nowego oprogramowania od podstaw. Środowisko to stanowi wspólną platformą dla wielu zaawansowanych narzędzi i bibliotek programistycznych do wspomagania skomplikowanych obliczeń matematycznych. System Sage pozwala na wykonywanie działań w sposób symboliczny oraz numeryczny. Całość środowiska została napisana w języku Python (głównie implementacja CPython) i umożliwia:

- obliczenia statystyczne - pakiet R oraz biblioteki NumPy,
- operacje symboliczne - program Maxima oraz biblioteka SymPy,
- symulacje numeryczne - biblioteka NumPy oraz pakiet Octave,
- bezpośrednio zastosowanie Python'a - terminal IPython,
- integrację obliczeń z tekstem - dokument \LaTeX oraz moduł SageTeX,
- pracę w środowisku rozproszonym - interfejs www na domenie www.cocalc.com.

Program składa się tylko z darmowych bibliotek i jest rozprowadzany na licencji GNU GPL. Oznacza to, możliwość całkowicie darmowego użytku do celów prywatnych oraz komercyjnych. Przedstawione zalety powodują, że środowisko SageMath stanowi ciekawą alternatywę dla płatnych programów takich jak Mathematica, MATLAB czy MathCad. Szczególnie wygodne jest jego zastosowanie przy pisaniu dłuższych tekstów o charakterze naukowym, ze względu na możliwość wykorzystania środowiska \LaTeX oraz integrację systemu CAS z edytorem tekstu.

3.2. Język skryptowy Python

Python jest wieloparadygmatowym językiem programowania wysokiego poziomu ogólnego przeznaczenia. Jednym z podstawowych założeń twórców było stworzenie języka o bardzo wysokiej czytelności kodu. Struktura programu stanowi jeden z elementów składni. Jest to wygodne, ponieważ zmusza programistę do dbania o zachowanie odpowiedniej przejrzystości kodu. Omawiany język wspiera koncepcje programowania strukturalnego, imperatywnego, obiektowego oraz w pewnym stopniu funkcyjnego. Ponadto Python charakteryzuje się dynamicznym typowaniem danych oraz automatycznym zarządzaniem pamięcią. Oznacza to, że obiekty są inicjowane w zależności od wprowadzanych danych. Przykładowo liczby z założenia inicjują obiekty liczbowe, pomimo że teoretycznie mogłyby konstruować obiekt typu znakowego. W większości praktycznych przypadków nie powoduje to błędów w kodzie.

Dynamiczne typowanie oraz automatyczne usuwanie niewykorzystywanych obiektów powoduje, że Python jest często stosowany jako język skryptowy. Cechą znamioną tego języka jest brak zapewnienia enkapsulacji właściwości obiektów w momencie deklaracji pól. Zapewnia on odpowiednie mechanizmy zapewniające hermetyzację, ale nie są one wymagane, tak jak ma to miejsce np. w języku C.

Python zapewnia swoją dużą uniwersalność przez możliwość zastosowania dodatkowych bibliotek. Ze względu na łatwość programowania w tym język zbiór dodatkowych modułów jest bardzo obszerny i pozwala zastosować Pythona niemalże w każdej dziedzinie. W obliczenia inżynierskich stosuje się głównie biblioteki NumPy, SymPy oraz Matplotlib, wchodzące w skład większego pakietu SciPy. Wykorzystanie środowiska SageMath oraz pakietu SageTeX pozwala na łatwe i efektywne osadzanie wyników skryptów Pythona w dokumentach *.tex.

3.3. Biblioteka NumPy

Biblioteka NumPy jest zbiorem procedur numerycznych napisanych w języku Python. Twórcą tego pakietu jest Travis Oliphant [[numpy`wikipedia](#)]. NumPy przede wszystkim dostarcza wygodnego interfejsu do pracy z macierzami. Oferuje składnie i funkcjonalność porównywalną z komercyjnymi rozwiązaniami takim jak Matlab, czy jego darmowy odpowiednik GNU Octave. Głównym zastosowaniem pakietu jest wspomaganie skomplikowanych obliczeń matematycznych i inżynierskich na dużych zbiorach danych.

Biblioteka dostarcza wiele powszechnie stosowanych procedur numerycznych. W skład pakietu wchodzi między innymi moduły do:

- obliczeń na przestrzeniach wektorowych,

- numerycznego rozwiązywania równań różniczkowych,
- poszukiwania pierwiastków zagadnień liniowych i nieliniowych,
- analizy sygnałów,
- transformacji Fouriera,
- operacji na dużych zbiorach danych,
- analizy obrazów.

Dokładną listę procedur oraz klas można znaleźć na stronie projektu [[numpy references](#)].

Zbiór NumPy pozwala na sprawne rozwiązywanie wielu praktycznych problemów inżynierskich. Podstawowa jej funkcjonalność dostarcza łatwy interfejs do tworzenia macierzy oraz wykonywania na nich operacji. Szczególnie często wykorzystywane są procedury `numpy.arange()`, `numpy.linspace()` pozwalające na generowanie macierzy oraz operatory `+`, `-`, `*` oraz `^` zastępujące standardowe implementacje działań dla języka Python (przeciążenie operatora).

W przypadku analizy problemów analizy sztywnościowej oraz dynamiki układów fizycznych NumPy dostarcza zbiory procedur `numpy.linalg.inv`, `lstinlinumpy.desolve`, `numpy.linalg.eig` oraz `numpy.fft` do odwracania macierzy, rozwiązywania równań różniczkowych, poszukiwania wartości i wektorów własnych oraz przetwarzania sygnałów.

Obliczenia inżynierskie bardzo często wymagają działań na macierzach oraz wektorach. Definiowanie macierzy z wykorzystaniem języka Python wymaga doładowanie odpowiednich bibliotek. Kod programu generującego wektor przedstawiający zakres liniowy jest następujący:

```
import numpy as np
np.linspace(2.0, 3.0, num=5)
```

Wynikiem programu jest wektor o postaci:

```
array([ 2. ,  2.2,  2.4,  2.6,  2.8])
```

Otrzymany wynik jest traktowany jako tablica. Założeniem programu było wygenerowanie wektora. W tym aspekcie środowisko NumPy jest bardzo zbliżone do komercyjnego oprogramowania Matlab, które wszystkie obiekty liczbowe traktuje jako macierze (w tym przypadku tablice). Wektory i skalary są reprezentowane jako odpowiednio macierze wierszowe oraz jednoelementowe.

W większości praktycznych przypadków taki zapis nie powoduje żadnych problemów interpretacyjnych. Jednocześnie znacząco upraszcza składnię.

3.4. Biblioteka Matplotlib

Biblioteka Matplotlib należąca do pakietu SciPy stanowi zbiór procedur pozwalających na graficzną reprezentację danych. Umożliwia wykonanie niemal dowolnego wykresu bazując na zestawie danych numerycznych. Wprowadzone środowisko jest napisane zgodnie z paradygmatem programowania obiektowego. Istotną zaletą pakietu jest interfejs PyLab. Jego celem jest upodobnienie procesu wykonywania wykresów w sposób możliwie najbliższy do stosowanego w środowisku MATLAB [matplotlib`wiki].

Wykorzystując środowisko Matplotlib możliwe jest wykonywanie różnego typu wykresów. Najczęściej wykorzystywane z nich to:

- wykresy dwuwymiarowe,
- wykresy trójwymiarowe,
- histogramy,
- wykresy z polami błędów,
- wykresy słupkowe.

Możliwe jest dodawanie do sceny dodatkowych grafik, opisów, wzór oraz dowolne formatowanie osi. Wyświetlane wartości mogą być reprezentowane w skali liniowej lub logarytmicznej. Ponadto każdy typ wykresu można wykreślić w postaci wielu pól na jednej scenie (tak zwane subplots).

Brak uniwersalnych wymagań dotyczących sposobu wyświetlania danych powoduje, że nie istnieje jedna metoda reprezentacji i opisu wartości znajdujących się na wykresie. W zależności od przeznaczenia i późniejszego zastosowania grafiki zalecenia odnośnie do formatowania mogą być inne. Środowisko Matplotlib umożliwia bardzo elastyczny wybór kroju, wielkości oraz rodzaju pisma. Wykorzystywane są do tego metody: `matplotlib.rcParams['text.usetex']`, `matplotlib.rcParams['text.latex.unicode']` i `matplotlib.rcParams['mathtext.fontset']`. Odpowiadają one za wykorzystanie środowiska \LaTeX oraz zastosowanie odpowiedniej czcionki do opisów osi.

W przypadku prac dyplomowych realizowanych w Politechnice Warszawskiej nie istnieją ścisłe wytyczne jak opracowywać wykresy zamieszczane w opracowaniach [rozporządzenie`pw`dyplomy]. Natomiast wymagania ogólne zalecają stosowanie bezszeryfowego kroju pisma w całej pracy. Zazwyczaj dodatkowo stosuje się pochyły krój szeryfowy dla oznaczenia symboli matematycznych.

Kod programu generującego wykres sformatowany w opisany sposób jest następujący:


```

import numpy as np

vector=np.array([ 2.0 ,  2.2,  2.4,  2.6,  2.8])

import matplotlib
matplotlib.rcParams['text.usetex'] = False
matplotlib.rcParams['text.latex.unicode'] = False
matplotlib.rcParams['mathtext.fontset'] = 'cm'
import matplotlib.pyplot as plt

plt.plot(vector)

plt.xlabel(r'$n_{[-]}$')
plt.ylabel(r'$\vector_{[m]}$')
plt.title('Wykres')

plt.show()

```

Zgodnie z oczekiwaniem, efektem działania programu jest wykres przedstawiający linię prostą. Metodą `plt.plot()` można nanieść na scenę dowolny zbiór danych. Jedynym wymaganiem jest numeryczna postać danych.

3.5. Pakiet SageTeX

Pakiet SageTeX jest pakietem dla systemu L^AT_EX umożliwiającym wykonywanie obliczeń bezpośrednio w opracowywanym dokumencie. Działania algebraiczne i przekształcenia są realizowane w środowisku SageMath jako integralna część dokumentu głównego `.tex`. Operacje matematyczne wyrażone są głównie w języku Python (patrz rozdział 3.1). Arkusz roboczy systemu Sage dołączany jest zbioru plików i wykonywany każdorazowo podczas przebudowywania dokumentu. Ze względów praktycznych warto ograniczać długość wykonywanego kodu. Przekłada się to na czas niezbędny do wygenerowania pliku wynikowego.

Moduł SageTeX dostarcza kilku środowisk/trybów wprowadzanie danych oraz wykonywania obliczeń. Wyniki działań można wstawić w trybie:

- `inline`, który pozwala na wykonanie obliczeń bezpośrednio w tekście. Wywoływany jest przy po-

mocy komendy `\sage{<kod programu>}` i zwraca bezpośrednio wzory w formacie \LaTeX gotowe do osadzenia w tekście;

- `sageplot`, gdzie środowisko SageMath wykonuje niezbędne obliczenia i automatycznie generuje wykres, a następnie osadza grafikę w tekście. Procedurę wywołuje się poleceniem `\sageplot{<kod programu>}`. Funkcja jest tożsama z wykorzystaniem makra `\includegraphics{.\\ściezka\do\nazwa_pliku.rozszerzenie}` i pozwala na zastosowanie wszystkich jej atrybutów;
- `sageblock`, który jest środowiskiem (z angielskiego `environment`) systemu \LaTeX . Wywoływane jest komendą `\begin{sageblock}`. Program wykonywany jest do wystąpienia frazy `\end{sageblock}`. Kod zawarty w bloku jest jednocześnie wykonywany i wyświetlany w dokumencie.
- `sageverbatim`, gdzie wykonywane obliczenia są tylko wyświetlane. Realizowane operacje umieszcza się w środowisku o tej samej nazwie, pomiędzy komendami `\begin{sageverbatim}` i `\end{sageverbatim}`;
- `sagesilent`, będący środowiskiem do rachunków nie umieszczanych w teście. Obliczenia są zapisywane w dodatkowym pliku z rozszerzeniem `.sage`. Kod zamieszcza się w pomiędzy poleceniami `\begin{sagesilent}` i `\end{sagesilent}`.

Możliwości trybów działania oferowanych przez pakiet Sage \TeX pozwalają na wykonywanie zróżnicowanych działań. Pozwala to zintegrować obliczenia z tekstem dla większości praktycznych przypadków, począwszy od prostych rachunków w tekście, a skończywszy na dużych obliczeniach z wykorzystaniem środowisk `sageblock` i `sagesilent`.

Zasada działania modułu Sage \TeX nie jest optymalna z punktu widzenia czasu niezbędnego do przebudowania dokumentu. Jak wspomniano na początku tego rozdziału, do wygenerowania dokumentu wynikowego niezbędne jest przeliczenie arkusza programu SageMath. Moduł implementuje funkcjonalność eliminowania zbędnych kompilacji, ale w praktyce nie zawsze się to sprawdza. Dlatego wygodnie jest skorzystać z niezależnego programu obliczeniowego (arkusza SageMath), z którego wyniki będą eksportowane do zbioru systemu \LaTeX . Importowanie zapisanych wyników odbywa się z wykorzystaniem makra `\sage{<kod SageMath>}` oraz procedury `load('nazwapliku.sobj')`. Ponadto odpowiada to praktyce, ponieważ w w większości projektów, obliczenia są wykonywane wcześniej niż opis przeprowadzonych prac. W ten sposób można wykorzystać dotychczasowe wyniki. W przypadku zależności matematycznych możliwe jest wywołanie procedury `\sage{load('nazwapliku.sobj')}` w środowisku `equation`. W ten sposób otrzymuje się numerowane równanie zawierające

wyniki obliczeń. Sam kod pozostaje bardzo wydajny, ponieważ polega tylko na odczytanie obiektu języka Python. Ostatecznie otrzymuje się zintegrowany z obliczeniami tekstowy.

$$\omega_1 = x^3 \tag{3.1}$$

Stosowanie pakietu SageTeX pozwala na bardziej wydajną pracę z zależnościami matematycznymi i wynikami prowadzonych obliczeń. Ma na to wpływ kilka aspektów. Przede wszystkim rozpatrywany pakiet pozwala na osadzanie skalowalnych zależności, które będą odpowiadać na pojawiające się zmiany w tekście i obliczeniach. Kolejnym aspektem jest uniknięcie konieczności przepisywania wzorów do środowiska L^AT_EX. Ostatecznie pakiet SageTeX pozwala na integrację dokumenty z dowolnym zewnętrznym arkuszem programu SageMath i wykorzystywać w bardzo wydajny sposób wykonane tam rachunki.

3.6. Eksportowanie wyników do dokumentu L^AT_EX

Elementy, które występują większości prac dyplomowych i przejściowych realizowanej na Politechnice Warszawskiej to zależności matematyczne, materiał graficzny oraz wykresy. W większości przypadków wzory oraz wykresy są wynikiem przeprowadzonych w pracy analiz. Zmiany obliczeń w trakcie realizacji tezy są zjawiskiem powszechnym. Dlatego warto rezultaty rachunków połączyć z tekstem w sposób dynamiczny. Środowisko SageMath dostarcza procedury (metody) `.save()` dla większości występujących tam obiektów. W przypadku zależności matematycznych zapis wywołuje się bezpośrednio funkcją `save(nazwa_pliku)`. Po jej wywołaniu w katalogu roboczy pojawia się plik o nazwie `nazwa_pliku.sobj`. Przy pomocy procedury `.load()` możliwe jest odczytanie zapisanego elementu. Pliki graficzne lub wykresy wygenerowane z pomocą biblioteki Matplotlib mogą być zapisywane przy pomocy metody `.savefig(nazwa_pliku.rozszerzenie)`. W ten sposób do kartoteki dodany zostanie plik graficzny o zadanym rozszerzeniu. Dalej można go wykorzystywać w standardowy sposób.

Osadzanie i automatyczne odświeżanie elementów w tekście zależy głównie od rozpatrywanego obiektu. Tak jak to opisano w rozdziale 3.5 zależności matematyczne najwygodniej zapisywać jako pliki `.sobj` i wczytywać je przy pomocy pakietu SageTeX.

Kod wyświetlający wzór oraz generujący numerację w środowisku `equation` jest następujący:

```
\begin{equation}
  \sage{load('nazwa_pliku.sobj')}
\end{equation}
```

Schematy oraz wykresy tworzone w arkuszu SageMath są zapisywane jako pliki graficzne. Zgodnie z opisem przedstawionym w rozdziale 2.4 materiał ilustracyjny dołącza się poleceniem `\includegraphic{. \sciezka\do\nazwa_pliku.rozszerzenie}`.

Program generujący pusty wykres, zmieniający się dynamicznie w trakcie zmian obliczeń jest następujący:

```
import time
import datetime

#variable which describes plot title - dynamic changing property
nazwa_wykresu = 'Data_□utworzenia_' + datetime.datetime.now()

import numpy as np

vector=np.array([ 2.0 ,  2.2,  2.4,  2.6,  2.8])

import matplotlib
matplotlib.rcParams['text.usetex'] = False
matplotlib.rcParams['text.latex.unicode'] = False
matplotlib.rcParams['mathtext.fontset'] = 'cm'
import matplotlib.pyplot as plt

plt.title(nazwa_wykresu)

plt.savefig('nazwa_pliku.rozszerzenie')
```

Kod dołączający utworzony plik graficzny ma postać:

```
\begin{figure}\label{fig:etykieta_wykresu}
  \includegraphic{. \sciezka\do\nazwa_pliku.rozszerzenie}
  \caption{Podpis rysunku}
\end{figure}
```

Przygotowane w ten sposób środowisko obliczeniowe SageMath oraz dokument \LaTeX pozwala na automatyzację procesu uaktualniania zależności matematycznych oraz grafiki.

Zaprezentowane rozwiązania znacząco usprawniają działania związane z opracowywaniem zagadnienia. Środowiko \LaTeX dostarcza gotowych rozwiązań do automatycznego generowanie spisów,

numeracji, bibliografii itp. Ponadto oprogramowanie to odciąża autora tekstu od żmudnego procesu formatowania. Szerokości tekstu, wielkość interlinii oraz liczba znaków w wierszu jest kontrolowana przez aplikację w taki sposób, aby efekt wizualny był jak najlepszy. Wykorzystanie pakietu SageTeX oraz środowisko SageMath połączenie arkusza obliczeniowego (programu napisanego w języku programowania Python) z opracowaniem. Przy odpowiednim sposobie eksportowanie wyników oraz ich wczytywania do dokumentu tekstowego jest to metoda efektywna. Ponadto zmniejsza wymagany na przebudowanie dokumentu czas. Materiał graficzny można skutecznie generować z wykorzystaniem biblioteki Matplotlib, która z interfejsem PyPlot pozwala na rysowanie wykresów w sposób znany z komercyjnego systemu MATLAB. Zastosowanie wszystkich przedstawionych rozwiązań skutkuje wygenerowaniem skalowalnego dokumentu, adaptującego się do częstych zmian jakie towarzyszą pisaniu prac przejściowych i dyplomowych.

4. Podsumowanie

Cel pracy dyplomowej został zrealizowany. Na wstępie przedstawiono wprowadzenie do tematyki, motywację podjęcia problematyki oraz metodykę rozwiązania. Dokonano przeglądu literatury światowej. Przedstawiono istniejące rozwiązania oraz porównano ich skuteczność w pisaniu prac dyplomowych. Opisano aktualny stan wiedzy oraz podkreślono słuszność podjęcia proponowanej tematyki dyplomu. W dalszej części pracy opisano jej strukturę, zastosowany język tekstu, narzędzia do jego edycji oraz możliwe do wykorzystania programy wspomagające obliczenia inżynierskie.

W kolejnym rozdziale przedstawiono zastosowanie oprogramowania \LaTeX . Przybliżono strukturę budowy dokumentu w tym: wykorzystania pakietów, metodyki tworzenia wzorów, tabel oraz rysunków. Opisano składnię umożliwiającą automatyczne generowanie spisów treści, tabel, rysunków i wzorów. Dodatkowo możliwe jest stworzenie bibliografii jako zbiór kolejnych pozycji cytowanych w docelowym dokumencie. Przedstawiono szereg przydatnych funkcji oprogramowania \LaTeX .

Kolejny rozdział opisywał możliwość zastosowania środowiska SageMath jako narzędzie wspomagania obliczeń inżynierskich. Oprogramowanie wykorzystuje m.in. język Python. Pozwala on na wykonywanie szeregu obliczeń począwszy od najprostszych działań matematycznych kończąc na złożonych symulacja oraz analizach widmowych co jest istotne ze względu na rozwiązywanie zadań z tematyki dynamiki złożonych układów. Przedstawiono wykorzystanie bibliotek NumPy oraz Matplotlib. Na koniec opisano możliwość sprzęgnięcia środowiska SageMath z oprogramowaniem \LaTeX . Taki zabieg pozwala na szybki sposób wykonywania obliczeń inżynierskich oraz tworzenia dokumentów tekstowych zawierających uzyskane wyniki w postaci liczbowej oraz graficznej.

Całość pracy podsumowano syntetycznymi wnioskami. Uzyskane wyniki pozwoliły zaproponować proces łatwego generowania dokumentów spełniających warunki stawiane pracom dyplomowym oraz artykułom naukowym.

Bibliografia

- [1] Dilip Datta. *LaTeX in 24 Hours: A Practical Guide for Scientific Writing*. 2017.
- [2] "Kurs Latex". W: (dostępne dnia 30.09.2018). URL: <http://latex-kurs.x25.pl/>.
- [3] Wikipedia. "Latex". W: (dostępne dnia 01.10.2018). URL: <https://pl.wikipedia.org/wiki/LaTeX>.
- [4] Rhett Allain. "Do rubber bands acts like springs?" W: (2012). URL: <https://www.wired.com/2012/08/do-rubber-bands-act-like-springs/>.
- [5] Joanna Karłowska-Pik Bartosz Ziemkiewicz. *LaTeX dla matematyków*. 2013.

Wykaz symboli i skrótów

computer π

Słownik terminów

computer is a programmable machine that receives input, stores and manipulates data, and provides output in a useful format. 37

π ratio of circumference of circle to its diameter. 21, 37

B	- indukcja magnetyczna w Teslach
l	- długość przewodu w metrach
v	- prędkość przecinania linii sił pola magnetycznego w kierunku prostopadłym w m/s
I	- natężenie prądu w Amperach
M	- moment obrotowy
φ	- strumień magnetyczny
R_t	- rezystancja twornika
I_t	- wartość prądu twornika
U	- napięcie zasilania
E	- siła elektromotoryczna
n	- prędkość obrotowa
φ_m	- strumień maksymalny
$(n_1 - n)$	- prędkość obrotowa względem wirnika, nazywana również poślizgiem
E_2	- Siła elektromotoryczna indukowana w jednej fazie uzwojenia nieruchomego wirnika
I_2	- prąd płynący w wirniku
ψ_2	- kąt przesunięcia fazowego między E_2 oraz I_2
M	- moment obrotowy wywierany na wirnik przez wirujące pole magnetyczne,
ω_1	- prędkość kątowa pola wirującego,
n_1	- prędkość obrotowa wirującego pola synchronicznego
c	- stała konstrukcyjna
β	- obrót linii działania gumy
r_1	- ramię pierwsze
r_2	- ramię drugie
F_w	- siła na gumie
F_v	- reakcja pionowa na drgania giętne
a	- odległość pionowa osi zaburzacza od osi wału
b	- odległość pozioma osi zaburzacza od osi wału
I_1	- moment bezwładności wirnika
I_2	- moment bezwładności zaburzacza
φ_1	- kąt obrotu wirnika
φ_2	- kąt obrotu zaburzacza
k_{zab}	- współczynnik sprężystości zaburzacza
$(l - l_0)$	- wydłużenie

Spis rysunków

2.1 Gotowe stanowisko	22
---------------------------------	----

Spis tabel

2.1 Wyniki eksperymentu	23
-----------------------------------	----

Spis załączników

1. Nazwa 1 Przykładowe wzory
2. Nazwa 2 załącznika
3. Nazwa 3 załącznika
4. Nazwa 4 załącznika
5. Nazwa 5 załącznika

Załączniki

Załącznik 4.1: Przykładowe wzory

Przykładowe wzory:

```
\begin{lstlisting}[language=Python]
\begin{equation}
f_2 = \frac{p \left( n_1 - n \right)}{60} = \frac{pn_1}{60} \frac{\left( n_1 - n \right)}{n_1}
\end{equation}
```

$$f_2 = \frac{p(n_1 - n)}{60} = \frac{pn_1}{60} \frac{(n_1 - n)}{n_1} \quad (4.1)$$

```
\begin{equation}v_i = \frac{dr_i}{dt} = \frac{dr_i}{\partial q_1} q_1 + \frac{dr_i}{\partial q_2} q_2 + \dots + \frac{dr_i}{\partial q_s} q_s + \frac{dr_i}{\partial t}
\end{equation}
```

$$v_i = \frac{dr_i}{dt} = \frac{dr_i}{\partial q_1} q_1 + \frac{dr_i}{\partial q_2} q_2 + \dots + \frac{dr_i}{\partial q_s} q_s + \frac{dr_i}{\partial t} \quad (4.2)$$

```
\begin{equation} \frac{d}{dt} \frac{\partial r_i}{\partial q_j} = \sum_{l=1}^s \frac{\partial^2 r_i}{\partial q_j \partial q_l} q_l + \frac{\partial^2 r_i}{\partial q_j \partial t}
\label{pierwsze_przekasz}
\end{equation}
```

$$\frac{d}{dt} \frac{\partial r_i}{\partial q_j} = \sum_{l=1}^s \frac{\partial^2 r_i}{\partial q_j \partial q_l} q_l + \frac{\partial^2 r_i}{\partial q_j \partial t} \quad (4.3)$$

```
\begin{equation} \frac{d}{dt} \sum_{i=1}^n \frac{\partial}{\partial q_j} \left( \frac{1}{2} m_i \dot{r}_i^2 \right) - \frac{\partial}{\partial q_j} \left( \sum_{i=1}^n \frac{1}{2} m_i \dot{r}_i^2 \right)
\end{equation}
```

$$\frac{d}{dt} \sum_{i=1}^n \frac{\partial}{\partial q_j} \left(\frac{1}{2} m_i r_i^2 \right) - \sum_{i=1}^n \frac{\partial}{\partial q_j} \left(\frac{1}{2} m_i r_i^2 \right) = \frac{d}{dt} \frac{\partial}{\partial q_j} \left(\sum_{i=1}^n \frac{1}{2} m_i r_i^2 \right) - \frac{\partial}{\partial q_j} \left(\sum_{i=1}^n \frac{1}{2} m_i r_i^2 \right) \quad (4.4)$$

`\begin{equation}q_{\{a\}}=T_{\{a\}} \ \delta_{\{aa\}}+T_{\{b\}} \ \delta_{\{ab\}}\end{equation}`

$$q_a = T_a \delta_{aa} + T_b \delta_{ab} \quad (4.5)$$

`\begin{equation}a_{\{n\}}=\frac{2}{T} \ \int_{\{0\}}^{\{T\}} F \ \left(t \ \right) \ \sin n \ \omega t \ \text{d}t`

$$a_n = \frac{2}{T} \int_0^T F(t) \sin n \omega t dt, \quad n = 1, 2, \dots \quad (4.6)$$

`\begin{equation}F \ \left(t \ \right) =\frac{1}{2i} \ \left[\sum_{\{n=1\}}^{\{\infty\}} \ \left(a_{\{n\}}+ib_{\{n\}} \ \right) \ e^{\{in \ \omega t\}}- \sum_{\{n=0\}}^{\{\infty\}} \ \left(a_{\{n\}}-ib_{\{n\}} \ \right) \ e^{-\{in \ \omega t\}} \ \right]`

$$F(t) = \frac{1}{2i} \left[\sum_{n=1}^{\infty} (a_n + ib_n) e^{in\omega t} - \sum_{n=0}^{\infty} (a_n - ib_n) e^{-in\omega t} \right] \quad (4.7)$$

`\begin{equation}h+\omega_0^2 h=e\Omega^2 \cos \Omega t+\frac{e\alpha_{\beta}}{2\beta^2 J} \left\{ \sin [(\beta+1)\Omega t+\eta_{\beta}]-\sin [\beta-1]\Omega t+\eta_{\beta} \right\}`

$$h + \omega_0^2 h = e\Omega^2 \cos \Omega t + \frac{e\alpha_{\beta}}{2\beta^2 J} \{ \sin [(\beta + 1)\Omega t + \eta_{\beta}] - \sin [\beta - 1]\Omega t + \eta_{\beta} \} \quad (4.8)$$

`\begin{equation}`

`\begin{bmatrix}`

`I_1 \ & \ 0 \\`

`0 \ & \ I_2`

`\end{bmatrix}`

`\begin{bmatrix}`

`\ddot{\phi}_1 \\`

`\ddot{\phi}_2`

`\end{bmatrix}`

`+`

`\begin{bmatrix}`

`\kappa + s \ & \ -\kappa \\`

`-\kappa \ & \ \kappa`

```

\end{bmatrix}
\begin{bmatrix}
\phi_1 \\
\phi_2
\end{bmatrix}
=
\begin{bmatrix}
M_n \\
0
\end{bmatrix}
\end{equation}

```

$$\begin{bmatrix} I_1 & 0 \\ 0 & I_2 \end{bmatrix} \begin{bmatrix} \ddot{\phi}_1 \\ \ddot{\phi}_2 \end{bmatrix} + \begin{bmatrix} \kappa + s & -\kappa \\ -\kappa & \kappa \end{bmatrix} \begin{bmatrix} \phi_1 \\ \phi_2 \end{bmatrix} = \begin{bmatrix} M_n \\ 0 \end{bmatrix} \tag{4.9}$$

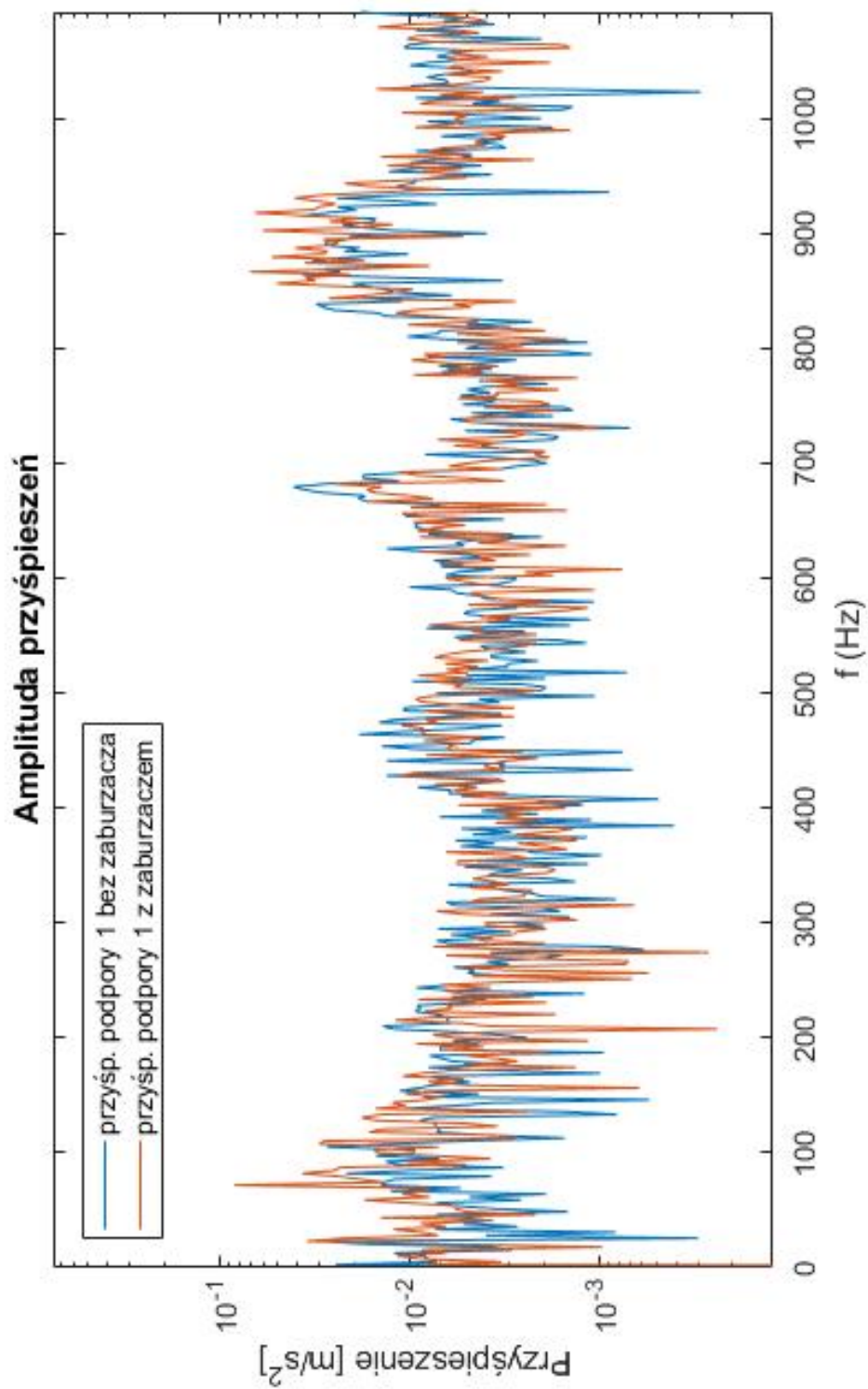
Listing 4.2: Nazwa drugiego załącznika

Tekst drugiego załącznika

Listing 4.3: Nazwa trzeciego załącznika

Tekst trzeciego załącznika

Listing 4.4: nazwa czwartego załącznika z rysunkiem



Listing 4.5: nazwa piątego załącznika z rysunkiem

