

**Akademia Górniczo-Hutnicza  
im. Stanisława Staszica w Krakowie**

---

Wydział Informatyki, Elektroniki i Telekomunikacji

KATEDRA INFORMATYKI



**PRACA MAGISTERSKA**

**MARTA RYŁKO, ANNA SKIBA**

**RÓWNOLEGŁE ALGORYTMY OPTYMALIZACJI TORU  
PRZEJAZDU W NARCIARSTWIE ALPEJSKIM**

PROMOTOR:  
dr inż. Roman Dębski

Kraków 2013

## **OŚWIADCZENIE AUTORA PRACY**

OŚWIADCZAM, ŚWIADOMY ODPOWIEDZIALNOŚCI KARNEJ ZA POŚWIADCZENIE NIEPRAWDY, ŻE NINIEJSZĄ PRACĘ DYPLOMOWĄ WYKONAŁEM OSOBIŚCIE I SAMODZIELNIE, I NIE KORZYSTAŁEM ZE ŹRÓDEŁ INNYCH NIŻ WYMNIENIONE W PRACY.

.....

PODPIS

**AGH**  
**University of Science and Technology in Krakow**

---

Faculty of Computer Science, Electronics and Telecommunication

DEPARTMENT OF COMPUTER SCIENCE



**MASTER OF SCIENCE THESIS**

**MARTA RYŁKO, ANNA SKIBA**

**PARALLEL ALGORITHMS FOR SKI-LINE OPTIMISATION IN  
ALPINE SKI RACING**

SUPERVISOR:  
Roman Dębski Ph.D

Krakow 2013

Serdecznie dziękuję ...

## Spis treści

|  |           |
|--|-----------|
| <b>1. Wstęp.....</b>                       | <b>6</b>  |
| 1.1. Cele pracy .....                      | 7         |
| 1.2. Zawartość pracy.....                  | 7         |
| <b>2. Wstęp teoretyczny .....</b>          | <b>8</b>  |
| 2.1. Struktura dokumentu .....             | 8         |
| 2.2. Kompilacja.....                       | 8         |
| 2.3. Narzędzia .....                       | 9         |
| 2.4. Przygotowanie dokumentu .....         | 9         |
| <b>3. Istniejące rozwiązania .....</b>     | <b>11</b> |
| 3.1. Struktura dokumentu .....             | 11        |
| 3.2. Kompilacja.....                       | 11        |
| 3.3. Narzędzia .....                       | 12        |
| 3.4. Przygotowanie dokumentu .....         | 12        |
| <b>4. Proponowane rozwiązanie.....</b>     | <b>14</b> |
| 4.1. Model narciarza i środowiska.....     | 14        |
| 4.2. Opis matematyczny modelu .....        | 14        |
| 4.3. Numeryczne rozwiązanie problemu ..... | 14        |
| 4.4. Optymalizacja toru przejazdu.....     | 14        |
| 4.5. Architektura systemu .....            | 14        |

# 1. Wstęp

Narciarstwo alpejskie to dyscyplina z długą historią. Rozwój sportowej wersji narciarstwa alpejskiego rozpoczął się w połowie XIX wieku, jednak nadal nie ma i prawdopodobnie nigdy nie będzie naukowej formuły opisującej tor po jakim należy się poruszać, aby zadaną trasę przejechać najszybciej. Ogromna ilość czynników, które wpływają na czas przejazdu znacznie utrudnia jej znalezienie. W sportowych dyscyplinach narciarstwa alpejskiego celem jest przejechanie w jak najkrótszym czasie wyznaczonej trasy od startu do mety, przejeżdżając przez wszystkie ustawione na trasie bramki - wymuszające skęty.

Problem, jakiego rozwiązania podejmujemy się w pracy, to problem optymalizacyjny rozwiązywany za pomocą symulacji komputerowej. Problem dotyczy znalezienia optymalnego toru przejazdu narciarza po trasie slalomu, który nakłada ograniczenia na ten tor w postaci bramek. Każda bramka ściśle narzuca, z której strony należy ją przejechać, a ominięcie chociaż jednej z nich powoduje dyskwalifikację zawodnika.

Zdefiniowany przez nas problem jest interdyscyplinarny - z pogranicza fizyki i informatyki. Do dobrego zrozumienia zjawisk zachodzących na stoku narciarskim cenne jest też posiadanie własnych doświadczeń z jazdy po trasach slalomu. Wymagania te powodują, że problem nie jest trywialny do rozwiązania i w celu badania go nieodłączne są osoby o różnych kompetencjach.

Obecnie nie udało nam się znaleźć publicznie dostępnych prac, które podchodziłyby do rozwiązania tego praktycznego problemu. Zdajemy sobie sprawę, że problem jest bardzo złożony i próby jego rozwiązania to tak naprawdę rozwiązanie uproszczone tego problemu. Dodatkowo, uwzględnić trzeba fakt, że wiele zmiennych występujących w równaniach wpływa na siebie nawzajem, powodując zmiany niekoniecznie widoczne natychmiast. Może to na przykład sprawiać, że niewielka zmiana dokonana na początku jazdy może mieć znaczący wpływ na ostateczny wynik, co znacznie utrudnia wszelką predykcję na temat wpływu zmian. Aby rozwiązać problem, stworzyliśmy fizyczny model narciarza - zamodelowany jako punkt materialny o konfigurowalnych parametrach, co umożliwia porównanie wyników np. dla zawodników o różnych masach. Potraktowanie narciarza jako punktu materialnego jest pierwszym z zastosowanych uproszczeń, które zdecydowałyśmy się przyjąć w naszym rozwiązaniu. Stok modelowany jest jako płaszczyzna o zadanym kącie nachylenia, na której za pomocą współrzędnych oznaczamy miejsce występowania bramek. Dużym wyzwaniem było dobranie przybliżenia trasy przejazdu, aby umożliwić wystarczająco łatwe obliczenia i jednocześnie nie tracąc zbyt wiele na dokładności oddania realnej trasy. Łamana, którą wybrałyśmy jako rozwiązanie spełniające obydwa te wymagania, jest wystarczająco dobrym przybliżeniem jeśli narzucimy na nią dodatkowe ograniczenia jak eliminacja ostrych kątów załamania.

Kluczową częścią naszego rozwiązania jest wykorzystanie algorytmu genetycznego do wybrania pewnego lokalnego optimum trasy, a następnie przeprowadzamy lokalną optymalizację celem wygładzenia znalezionej trasy. Aby przyspieszyć obliczenia, zastosowałyśmy architekturę opartą o rozproszonych klientów wykonujących obliczenia i raportujących do głównego serwera. Na podstawie zebranych danych od klientów, serwer jest w stanie dostarczyć rozwiązanie szybciej oraz można mieć większą pewność, iż jest ono jeśli nie optymalne, to bardzo bliskie optymalnego. Obliczenia wykonywane są w środowisku przeglądarek internetowych w języku JavaScript.

Otrzymane rozwiązanie może mieć zastosowanie nie tylko w celu znajdowania optymalnej trasy przejazdu po zadanym slalomie. Przykładem może być wsparcie dla trenerów ustawiających takie slalomy w postaci aplikacji podpowiadającej gdzie ustawić kolejną bramkę, aby nie było problemów z jej przejechaniem. Dodatkowo, dokładając moduł wyliczający naprężenia i siły działające na stawy kolanowe, można by zredukować negatywny wpływ niefortunnie ustawionych bramek, powodujących wyjątkowe przeciążenia w kolanach, wykrywając to i przedstawiając bramki.

## 1.1. Cele pracy

Celem poniższej pracy jest zapoznanie studentów z systemem  $\text{\LaTeX}$  w zakresie umożliwiającym im samodzielne, profesjonalne złożenie pracy dyplomowej w systemie  $\text{\LaTeX}$ .

## 1.2. Zawartość pracy

W rozdziale ?? przedstawiono podstawowe informacje dotyczące struktury dokumentów w  $\text{\LaTeX}$ u. *Alvis* [3] jest językiem

## 2. Wstęp teoretyczny

W rozdziale tym przedstawiono informacje .

### 2.1. Struktura dokumentu

Plik  $\LaTeX$ owy jest plikiem tekstowym, który oprócz tekstu zawiera polecenia formatujące ten tekst (analogicznie do języka HTML). Plik składa się z dwóch części:

1. Preambuły – określającej klasę dokumentu oraz zawierającej m.in. polecenia dołączającej dodatkowe pakiety;
2. Części głównej – zawierającej zasadniczą treść dokumentu.

```
\documentclass[a4paper,12pt]{article}          % preambuła
\usepackage[polish]{babel}
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{times}

\begin{document}                                % część główna

\section{Sztuczne życie}

% treść

\end{document}
```

Nie ma żadnych przeciwwskazań do tworzenia dokumentów w  $\LaTeX$ u w języku polskim. Plik źródłowy jest zwykłym plikiem tekstowym i do jego przygotowania można użyć dowolnego edytora tekstów, a polskie znaki wprowadzać używając prawego klawisza `Alt`. Jeżeli po kompilacji dokumentu polskie znaki nie są wyświetlane poprawnie, to na 95% źle określono sposób kodowania znaków (należy zmienić opcje wykorzystywanych pakietów).

### 2.2. Kompilacja

Założmy, że przygotowany przez nas dokument zapisany jest w pliku `test.tex`. Kolejno wykonane poniższe polecenia (pod warunkiem, że w pierwszym przypadku nie wykryto błędów i kompilacja zakończyła się sukcesem) pozwalają uzyskać nasz dokument w formacie pdf:



```
latex test.tex
dvips test.dvi -o test.ps
ps2pdf test.ps
```

lub za pomocą PDF $\LaTeX$ :

```
pdflatex test.tex
```

Przy pierwszej kompilacji po zmianie tekstu, dodaniu nowych etykiet itp.,  $\LaTeX$  tworzy sobie spis rozdziałów, obrazków, tabel itp., a dopiero przy następnej kompilacji korzysta z tych informacji.

W pierwszym przypadku rysunki powinny być przygotowane w formacie eps, a w drugim w formacie pdf. Ponadto, jeżeli używamy polecenia `pdflatex test.tex` można wstawiać grafikę bitową (np. w formacie jpg).

## 2.3. Narzędzia

Do przygotowania pliku źródłowego może zostać wykorzystany dowolny edytor tekstowy. Niektóre edytory, np. Emacs, mają wbudowane moduły ułatwiające składanie tekstów w  $\LaTeX$ u (kolorowanie składni, skrypty kompilacji, itp.).

Jednym z bardziej znanych środowisk do składania dokumentów  $\LaTeX$ a jest *Kile*. Aplikacja dostępna jest dla środowiska KDE począwszy od wersji 2. Zawiera edytor z podświetlaną składnią, zestawy poleceń  $\LaTeX$ a, zestawy symboli matematycznych, kreatory tabel, macierzy, skrypty kompilujące i konwertujące podpięte są do poleceń w menu aplikacji (i pasków narzędziowych), dostępne jest sprawdzanie pisowni, edytor obsługuje projekty (tzn. dokumenty składające się z wielu plików), umożliwia przygotowanie i zarządzanie bibliografią, itp.

Na stronie <http://kile.sourceforge.net/screenshots.php> zamieszczono kilkanaście zrzutów ekranu środowiska *Kile*, które warto przejrzeć, by wstępnie zapoznać się z możliwościami programu.

Bardzo dobrym środowiskiem jest również edytor gEdit z wtyczką obsługującą  $\LaTeX$ a. Jest to standardowy edytor środowiska Gnome. Po instalacji wtyczki obsługującej  $\LaTeX$ a, edytor nie ustępuje funkcjonalnościom środowisku *Kile*, a jest zdecydowanie szybszy w działaniu. Lista dostępnych wtyczek dla tego edytora znajduje się pod adresem <http://live.gnome.org/Gedit/Plugins>. Inne polecane wtyczki to:

- Edit shortcuts – definiowanie własnych klawiszy skrótów;
- Line Tools – dodatkowe operacje na liniach tekstu;
- Multi-edit – możliwość jednoczesnej edycji w wielu miejscach tekstu;
- Zoom – zmiana wielkości czcionki edytora z użyciem rolki myszy;
- Split View – możliwość podziału okna edytora na 2 części.

## 2.4. Przygotowanie dokumentu

Plik źródłowy  $\LaTeX$ a jest zwykłym plikiem tekstowym. Przygotowując plik źródłowy warto wiedzieć o kilku szczegółach:

- Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja. Aby uzyskać *twardą spację*, zamiast znaku spacji należy użyć znaku *tyldy*.

- Znakiem końca akapitu jest pusta linia (ilość pustych linii nie ma znaczenia), a nie znaki przejścia do nowej linii.
- $\text{\LaTeX}$  sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.

## 3. Istniejące rozwiązania

W rozdziale tym przedstawiono informacje .

### 3.1. Struktura dokumentu

Plik  $\text{\LaTeX}$ owy jest plikiem tekstowym, który oprócz tekstu zawiera polecenia formatujące ten tekst (analogicznie do języka HTML). Plik składa się z dwóch części:

1. Preambuły – określającej klasę dokumentu oraz zawierającej m.in. polecenia dołączającej dodatkowe pakiety;
2. Części głównej – zawierającej zasadniczą treść dokumentu.

```
\documentclass[a4paper,12pt]{article}           % preambuła
\usepackage[polish]{babel}
\usepackage[latin2]{inputenc}
\usepackage[T1]{fontenc}
\usepackage{times}

\begin{document}                                % część główna

\section{Sztuczne życie}

% treść

\end{document}
```

Nie ma żadnych przeciwwskazań do tworzenia dokumentów w  $\text{\LaTeX}$ u w języku polskim. Plik źródłowy jest zwykłym plikiem tekstowym i do jego przygotowania można użyć dowolnego edytora tekstów, a polskie znaki wprowadzać używając prawego klawisza `Alt`. Jeżeli po kompilacji dokumentu polskie znaki nie są wyświetlane poprawnie, to na 95% źle określono sposób kodowania znaków (należy zmienić opcje wykorzystywanych pakietów).

### 3.2. Kompilacja

Załóżmy, że przygotowany przez nas dokument zapisany jest w pliku `test.tex`. Kolejno wykonane poniższe polecenia (pod warunkiem, że w pierwszym przypadku nie wykryto błędów i kompilacja zakończyła się sukcesem) pozwalają uzyskać nasz dokument w formacie pdf:

```
latex test.tex
dvips test.dvi -o test.ps
ps2pdf test.ps
```

lub za pomocą PDF $\LaTeX$ :

```
pdflatex test.tex
```

Przy pierwszej kompilacji po zmianie tekstu, dodaniu nowych etykiet itp.,  $\LaTeX$  tworzy sobie spis rozdziałów, obrazków, tabel itp., a dopiero przy następnej kompilacji korzysta z tych informacji.

W pierwszym przypadku rysunki powinny być przygotowane w formacie eps, a w drugim w formacie pdf. Ponadto, jeżeli używamy polecenia `pdflatex test.tex` można wstawiać grafikę bitową (np. w formacie jpg).

### 3.3. Narzędzia

Do przygotowania pliku źródłowego może zostać wykorzystany dowolny edytor tekstowy. Niektóre edytory, np. Emacs, mają wbudowane moduły ułatwiające składanie tekstów w  $\LaTeX$ u (kolorowanie składni, skrypty kompilacji, itp.).

Jednym z bardziej znanych środowisk do składania dokumentów  $\LaTeX$ a jest *Kile*. Aplikacja dostępna jest dla środowiska KDE począwszy od wersji 2. Zawiera edytor z podświetlaną składnią, zestawy poleceń  $\LaTeX$ a, zestawy symboli matematycznych, kreatory tabel, macierzy, skrypty kompilujące i konwertujące podpięte są do poleceń w menu aplikacji (i pasków narzędziowych), dostępne jest sprawdzanie pisowni, edytor obsługuje projekty (tzn. dokumenty składające się z wielu plików), umożliwia przygotowanie i zarządzanie bibliografią, itp.

Na stronie <http://kile.sourceforge.net/screenshots.php> zamieszczono kilkanaście zrzutów ekranu środowiska *Kile*, które warto przejrzeć, by wstępnie zapoznać się z możliwościami programu.

Bardzo dobrym środowiskiem jest również edytor gEdit z wtyczką obsługującą  $\LaTeX$ a. Jest to standardowy edytor środowiska Gnome. Po instalacji wtyczki obsługującej  $\LaTeX$ a, edytor nie ustępuje funkcjonalnościom środowisku *Kile*, a jest zdecydowanie szybszy w działaniu. Lista dostępnych wtyczek dla tego edytora znajduje się pod adresem <http://live.gnome.org/Gedit/Plugins>. Inne polecane wtyczki to:

- Edit shortcuts – definiowanie własnych klawiszy skrótów;
- Line Tools – dodatkowe operacje na liniach tekstu;
- Multi-edit – możliwość jednoczesnej edycji w wielu miejscach tekstu;
- Zoom – zmiana wielkości czcionki edytora z użyciem rolki myszy;
- Split View – możliwość podziału okna edytora na 2 części.

### 3.4. Przygotowanie dokumentu

Plik źródłowy  $\LaTeX$ a jest zwykłym plikiem tekstowym. Przygotowując plik źródłowy warto wiedzieć o kilku szczegółach:

- Poszczególne słowa oddzielamy spacjami, przy czym ilość spacji nie ma znaczenia. Po kompilacji wielokrotne spacje i tak będą wyglądały jak pojedyncza spacja. Aby uzyskać *twardą spację*, zamiast znaku spacji należy użyć znaku *tyldy*.

- Znakiem końca akapitu jest pusta linia (ilość pustych linii nie ma znaczenia), a nie znaki przejścia do nowej linii.
- $\text{\LaTeX}$  sam formatuje tekst. **Nie starajmy się go poprawiać**, chyba, że naprawdę wiemy co robimy.

## **4. Proponowane rozwiązanie**

W rozdziale tym przedstawiono informacje .

### **4.1. Model narciarza i środowiska**

### **4.2. Opis matematyczny modelu**

### **4.3. Numeryczne rozwiązanie problemu**

### **4.4. Optymalizacja toru przejazdu**

### **4.5. Architektura systemu**

## Bibliografia

- [1] A. DILLER, *LaTeX wiersz po wierszu*, Wydawnictwo Helion, Gliwice, 2000.
- [2] L. LAMPORT, *LaTeX system przygotowywania dokumentów*, Wydawnictwo Ariel, Krakow, 1992.
- [3] M. SZPYRKA, *On Line Alvis Manual*, AGH University of Science and Technology, 2011,  
<http://fm.ia.agh.edu.pl/alvis:manual>.