

POLITECHNIKA WARSZAWSKA

WYDZIAŁ ELEKTRYCZNY

Rozprawa doktorska

mgr inż. Łukasz Izdebski

Interpolacja trajektorii w animacji

Promotor

dr hab. inż. Dariusz Sawicki, prof. uczelni

Promotor pomocniczy

dr Ryszard Kopiecki

WARSZAWA, 2019

*Pragnę złożyć wyrazy wdzięczności mojemu promotorowi doktorowi habilitowanemu **Dariuszowi Sawickiemu** oraz doktorowi **Ryszardowi Kopieckiemu** za wsparcie naukowe, merytoryczne ukierunkowanie pracy i czas poświęcony przy jej realizacji. Cierpliwość i wyrozumiałość okazywane w trakcie przekazywania uwag oraz wskazówek nie do przecenienia sprawiły, iż nasza współpraca zaowocowała niniejszą rozprawą.*

*Dziękuję mojej **Źonie**, bez której niniejsza rozprawa by nie powstała oraz wszystkim bliskim i przyjaciołom, którzy wspierali moje działania.*

Streszczenie

Niniejsza rozprawa doktorska podejmuje tematykę związaną z animacją w szczególności animacją komputerową. Krzywe Béziera trzeciego stopnia są powszechnie wykorzystywane jako funkcje interpolujące w metodzie klatek kluczowych. Opracowana przez autora aproksymacja krzywych Béziera przy użyciu wielomianów trzeciego stopnia pozwala wyeliminować duży narzut obliczeniowy oraz narzut pamięciowy w interpolacji parametrów, przy użyciu wspomnianych krzywych, w aplikacjach czasu rzeczywistego.

Praca rozpoczyna się od próby zdefiniowania pojęcia animacji, oraz omówienia metody klatek kluczowych w animacji komputerowej. Następnie w pracy opisane są różnego typu funkcje interpolujące oraz zarysowany został problem wydajnościowy narzucony przez interpolację przy użyciu krzywych Béziera. W dalszej części zdefiniowano cel, tezę i zakres pracy oraz sformułowano zadania, których realizacja była niezbędna do wykazania słuszności postawionej tezy. Merytoryczna część pracy rozpoczyna się od przedstawienia krzywej Béziera trzeciego stopnia jako wykres funkcji. Rozważane są ograniczenia na parametry tej krzywej, pozwalające wykorzystywać ją jako poprawną funkcję interpolującą. Następnie wykorzystując wybraną przestrzeń unormowaną zdefiniowano funkcję celu dla zadania postawionego w pracy. Pozwala to wyznaczyć współczynniki wielomianów aproksymujących. Dodatkowo przedstawiono warunki konieczne oraz warunki opcjonalne, które narzucają na funkcję celu ograniczenia, pozwalające udostępnić użytkownikom możliwość konfiguracji zaproponowanej metody w poszukiwaniu odpowiedniej aproksymacji. Zaproponowana w pracy metoda została zaimplementowana a następnie poddana testom. Przeprowadzone eksperymenty wykazały poprawność zaproponowanego rozwiązania, a także pozwoliły wykazać efektywność nowej metody w różnych zastosowaniach. W zakończeniu pracy podsumowano uzyskane rezultaty, przedstawiono osiągnięcia autora oraz podkreślono słuszność postawionej w pracy tezy.

Słowa kluczowe: animacja, metoda klatek kluczowych, krzywa Béziera, funkcja sklejana, przestrzeń unormowana, przestrzeń Sobolewa

Abstract

This Ph.D. thesis describes issues related to animation, in particular computer animation. Cubic Bézier curves are commonly used as interpolating functions in the keyframes animation method. An approximation of Bézier curves developed by the author of this thesis using cubic polynomials allows to eliminate memory and performance overhead for real-time applications.

The thesis begins with the attempt to define a concept of animation and discuss the keyframes method in computer animation. Then, the thesis describes various types of interpolating functions and outlines the performance problem imposed by interpolation using Bézier curves. Next author formulated thesis and tasks which help to prove the rightness of the thesis. Main part of the thesis begins with the way how to present cubic Bézier curve as a function graph and describe the limits on parameters allowing to use Bézier curve as a proper interpolating function. Then, the thesis sets normed space which is used to define objective function for realization of the task defined in this work. This allows to determine the coefficients of approximate cubic polynomials. In addition, the necessary and optional constraint conditions are defined, which impose on the objective function, thus allowing users to configure proposed method in search of an appropriate approximation. Then proposed method was implemented and then tested. The experiments showed the correctness of the proposed solution and also allowed to demonstrate the effectiveness of the new method in various applications. At the end, author summarizes the obtained results and proves the validity of the thesis.

Keywords: animation, keyframe animation, Bézier curve, normed space, spline, Sobolev space

Spis treści

1	Wstęp	11
1.1	Próba zdefiniowania pojęcia animacji	11
1.2	Dynamika animacji	13
1.3	Metoda klatek kluczowych w animacji komputerowej	15
1.4	Właściwości krzywych interpolacyjnych	18
1.5	Reprezentacja krzywych interpolacyjnych w postaci jednej krzywej Béziera trzeciego stopnia	23
1.6	Funkcje interpolacyjne w praktyce	24
1.6.1	Aplikacje generujące obrazy komputerowe pracujące w trybie offline	25
1.6.2	Aplikacje czasu rzeczywistego	28
1.7	Wnioski	30
2	Cel, teza i zakres pracy	32
3	Krzywa Béziera	35
3.1	Krzywa Béziera jako wykres funkcji	35
3.1.1	Interpretacja krzywej Béziera w metodzie klatek kluczowych	38
3.1.2	Funkcja odwrotna do wielomianu co najwyżej trzeciego stopnia	41
3.1.3	Funkcja $C(x)$	45
4	Nowy sposób aproksymacji krzywych Béziera trzeciego stopnia	49
4.1	Opis metody	49
4.2	Warunki ciągłości w węzłach	50
4.2.1	Warunki konieczne	50
4.2.2	Warunki opcjonalne	51
4.3	Przestrzenie unormowane. Normy całkowite	52
4.3.1	Norma supremum	53
4.3.2	Przestrzenie L^p	53
4.3.3	Przestrzenie Sobolewa	54
4.3.4	Funkcja celu	55

4.4	Proponowana metoda wyznaczania współczynników wielomianów	56
4.4.1	Postać macierzowa równania	56
4.4.2	Warunki ciągłości w postaci macierzy rozszerzonej	60
4.5	Ekstrema warunkowe funkcji różniczkowalnych	62
4.5.1	Metoda mnożników Lagrange’a	62
4.6	Możliwe rozszerzenie metody o aproksymacje wielomianami wyższych stopni	64
4.7	Problem wyznaczenia optymalnych węzłów	65
4.8	Wnioski	67
5	Koncepcja sprawdzania poprawności rozwiązania	68
5.1	Wstępne informacje potrzebne do analizy proponowanej metody	68
5.1.1	Postać znormalizowana krzywej Béziara trzeciego stopnia	68
5.1.2	Ograniczenia dla postaci znormalizowanej	69
5.1.3	Krzywe Béziara poddane testom	71
5.2	Błąd aproksymacji a wizualny odbiór aproksymowanej trajektorii	73
5.2.1	Percepcja ruchu	74
5.2.2	Próba wyznaczenia wartości granicznej błędu aproksymacji	75
5.2.3	Wnioski	93
6	Testy oraz analiza rozwiązania	95
6.1	Błąd aproksymacji w normie $W^{1,2}$ a błąd w normie supremum	95
6.2	Testy ze względu na liczbę wielomianów aproksymujących funkcję wejściową $C(x)$	96
6.2.1	Porównanie błędów aproksymacji przy podziale równomiernym do podziału nierównomiernego metodą siłową	102
6.3	Testy błędów aproksymacji w warunkach ciągłości w węzłach	104
6.4	Testy błędów aproksymacji ze względu na funkcję wewnętrzną i zewnętrzną . .	110
6.4.1	Testy dla błędów aproksymacji ze względu na funkcję wewnętrzną . .	110
6.4.2	Testy dla błędów aproksymacji ze względu na funkcję zewnętrzną . . .	115
6.5	Rozszerzenie aproksymacji o wielomiany wyższych stopni	118
6.6	Porównanie wyników prezentowanej metody z wynikami interpolacji funkcjami sklejanymi	121
6.7	Postać znormalizowana a błąd aproksymacji	125
6.8	Testy pamięciowe	127
6.9	Testy wydajnościowe	130
7	Podsumowanie	133

A Historia animacji	138
A.1 Historia animacji klasycznej	138
A.2 Historia animacji komputerowej	151
B Techniki i metody animacji	155
C Rodzaje animacji komputerowej	157
D Rodzaje funkcji interpolacyjnych	159
E Próbkowanie	167
F Nierówność Sobolewa dla funkcji określonych na odcinku	172
F.1 Definicje	172
F.2 Podstawowe nierówności	173
Bibliografia	175

Rozdział 1

Wstęp

1.1 Próba zdefiniowania pojęcia animacji

Można zaryzykować stwierdzenie, że powszechność i wielokulturowość filmów animowanych sprawia, że niemal każdy człowiek na świecie zapytany o to, czy wie, co to jest „animacja”, odpowie twierdząco. Jednocześnie dzięki temu, że filmy animowane powstawały i powstają na całym świecie, może okazać się, że w różnych kręgach kulturowych, a nawet dla każdej osoby, definicja ta może być nieco inna.

Definicja animacji może być związana z lokalnymi nurtami tej sztuki. Dla przykładu: w Japonii animacja kojarzy się z filmami zwanymi przez resztę świata anime - animacją, która powstała na bazie komiksów „manga”. Są to animacje rysowane odręcznie [18], klatka po klatce, pełne estetyki, symboli i skrótów myślowych zakorzenionych w kulturze japońskiej. Z kolei w Polsce wielką popularnością cieszyły się przez lata produkcje studia Se-Ma-For. Kolejne pokolenia wychowywały się na bajkach o kocie Filemonie czy wróblu Ćwirku. Specjalizacją studia były animacje lalek - takie jak w bajkach o Misiu Uszatku, ekranizacji serii o Mumin-kach oraz bajki „Plastusiowy pamiętnik”. Animacja lalek, kukielki i wszelkie formy pośrednie rozpowszechnione były zarówno w Polsce jak i w niemal całej Europie, a jednocześnie niezwykle rzadko pojawiały się w animacjach tworzonych w Stanach Zjednoczonych. W efekcie tych rozbieżności spojrzenie na zagadnienie animacji jest różne w zależności od części świata.

Wskutek rozwoju nowych technik animacji to, z czym kojarzy się animacja, będzie także różnie opisywane w zależności od wieku odbiorcy. W Stanach Zjednoczonych osobom dorosłym animacja zapewne nieodłącznie kojarzy się z filmami, które powstawały w wielkich studiach Walt Disney Pictures czy Hanna-Barbera. Młodsze pokolenia wychowywały się na animacjach komputerowych studia Pixar.

Próby zdefiniowania pojęcia animacji były podejmowane wiele razy. Nie jest to proste zadanie, gdyż animacja, jak niemal każda dziedzina sztuki, wymyka się precyzyjnym opisom. Dodatkowym utrudnieniem jest różnorodność metod i technik wykorzystywanych w tworzeniu

filmów animowanych oraz ciągły rozwój i powstawanie nowych form. W Encyklopedii PWN wydanej w roku 1999 definicja animacji jest następująca:

„ANIMACJA [łac.], dokonywanie serii pojedynczych zdjęć rysunków, kukielek, wycinanek w różnych fazach ruchu, które przy wyświetlaniu w sposób ciągły dają na ekranie efekt ożywienia martwych kształtów”. [45]

Jak widać ta definicja jest ściśle związana z nurtem europejskim, w którym dużą rolę odgrywały animacje kukielkowe i im podobne. Już po niecałych 20 latach jest to jednak definicja nieaktualna. Opisana w tej definicji metoda: „*dokonywanie serii pojedynczych zdjęć*” wyklucza całkowicie animacje komputerowe.

W definicji animacji musi zostać uwzględnione szerokie spektrum technik i wielorakość przeznaczenia.

W roku 1987 członkowie Międzynarodowego Stowarzyszenia Filmu Animowanego zaproponowali następującą definicję:

„Sztuką animacji jest kreacja ruchomych obrazów przy użyciu różnorodnych technik z wyjątkiem reprezentacji akcji na żywo”.

Ta definicja nie zadowoliła jednak wszystkich artystów, gdyż wciąż okazała się zbyt ogólną. Jedną z przyczyn sporów o definicję jest fakt, że rozumienie postrzegania animacji jest ściśle związane z opisanym wcześniej kręgiem kulturowym odbiorcy, każdy widz sam intuicyjnie wie, czym jest animacja.

Zaledwie rok później, w 1988 roku, na ekrany kin wszedł (i stał się hitem) film „Kto wrobił królika Rogera”. Film zaliczany jest do kategorii filmów animowanych, chociaż łączy animowane postacie z rzeczywistym światem i grą aktorów.

Czy więc przy tak szerokiej gamie narzędzi i wielu możliwościach łączenia animacji z innymi sztukami możliwe jest precyzyjne jej zdefiniowanie? Wszystkie rozważania na temat animacji wiążą jeden wspólny wątek, jakim jest jej twórczy charakter.

Próbując określić definicję animacji warto zwrócić uwagę na etymologię i cechy wyróżniające tę sztukę spośród innych. Animacja posiada wyjątkową cechę, jaką jest wytwarzanie iluzji ożywiania. W języku łacińskim „anima” oznacza „dusza” lub „oddech”. Zarówno dusza jak i oddychanie wiążą się z życiem, zatem animacja to ożywianie, dawanie duszy przedmiotom, zwierzętom lub nawet całkowicie fikcyjnym twórcom. To wyróżnia animację na tle innych sztuk - pozwala ona „tchnąć życie” w statyczne obrazy. A z życiem od razu wiąże się główny jego aspekt, czyli ruch.

Animacja jest sztuką interdyscyplinarną, ponieważ korzysta z elementów malarstwa, rysunku, rzeźby, modelarstwa, tańca, technik komputerowych, ale także historii, psychologii i socjologii. Wytworzyła swój własny, swoisty język, w którym to, co do tej pory było nieosiągalne,

staje się możliwe. Animacja pozwala nam dojść do punktu, w którym jedynym ograniczeniem jest ludzka wyobraźnia.

1.2 Dynamika animacji

Aby obiekt był w ruchu (wywoływał takie wrażenie u widza), musi zmieniać, przykładowo, swoją pozycję w czasie. Jeżeli czas płynie, a pozycja obiektu nie ulega zmianie, obiekt postrzegany jest jako nieruchomy. Relacja między upływem czasu a postępującą zmianą parametru jest podstawą każdej sztuki opartej na czasie, takiej jak muzyka, taniec czy film (animacja). Dzięki temu, że artysta ma możliwość kontrolowania wielkości zmian oraz ich prędkości pomiędzy klatkami, pozycji, orientacji itp., własności obiektów powodują to że animowane obiekty wywołują u odbiorcy iluzję posiadania życia.

Aby to osiągać, animator musi opanować podstawowe zasady animacji, związane z dwoma parametrami, zwanymi wycuciem czasu oraz wycuciem odstępu (*ang. timing and spacing*). Zasady te, mimo że są bardzo łatwe do zrozumienia, prawdopodobnie należą do najtrudniejszych w poprawnym wykonaniu.

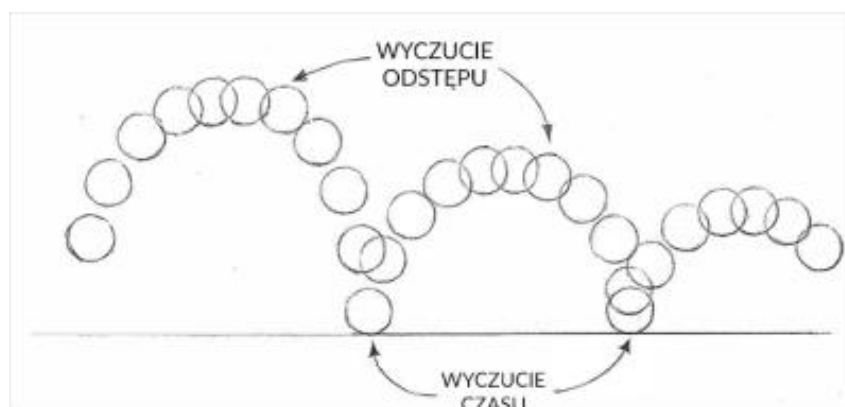
Wycucie czasu (*ang. timing*), jak sama nazwa wskazuje, jest to pojęcie odnoszące się w animacji do czasu, szybkości lub tempa. Określane jest na podstawie liczby obrazów (ramek) potrzebnych do wykonania danej akcji. Można tu wskazać na bardzo prostą zależność - im więcej klatek animacji obiekt potrzebuje do wykonania czynności, tym wolniejszą jest akcja, natomiast im mniej, tym szybszą.

Wycucie odstępu (*ang. spacing*) jest pojęciem określającym zmianę animowanego parametru w poszczególnych klatkach animacji, np. jak bardzo zmieniała się pozycja animowanego obiektu w czasie. Podobnie jak w przypadku wycucia czasu, również tu występuje zależność - czym większa zmiana pozycji, tym szybsza akcja, a czym mniejsza zmiana pozycji, tym akcja wolniejsza.

Zależności między wycuciem czasu oraz odstępu można przedstawić na podstawie przykładu odbijającej się piłki. Jeżeli została rzucona piłka (np. tenisowa, golfowa, siatkowa czy też kule do kręgli) posiadająca właściwości fizyczne, takie jak wielkość, sprężystość oraz wagę, z pewnej wysokości, to w zależności od rodzaju piłki będzie się ona inaczej zachowywać przy kontakcie z podłożem. Piłka siatkowa odbije się mocno i zanim się całkowicie zatrzyma, wykona kilkanaście odbić. Natomiast kula do kręgli wykona jedno odbicie i pozostanie już na podłożu.

Momenty styku każdego rodzaju piłki z podłożem wyznaczać będą rytm, który jest pomocny w ustaleniu szybkości oraz okresu odbijania się piłek. Wykonana tylko na podstawie *timing* animacja będzie wyglądała sztucznie, taki ruch nie będzie przypominał ruchu realnego odpowiednika. Aby urealnić ten ruch potrzebne jest zastosowanie odpowiedniego odstępu. W tym celu należy wziąć pod uwagę zasady dynamiki Newtona, zasadę bezwładności piłki, zasadę zachowania

wania pędu oraz prawo powszechnego ciężenia. Stopień, w jakim te niewidzialne siły działają na piłkę powoduje, że odbija się ona we właściwy sobie sposób (ruch zależy tylko od właściwości fizycznych samej piłki). Ruch przyspiesza, gdy piłka opada na ziemię, zwalnia zaś w miarę zbliżania się do momentu osiągnięcia maksymalnej wysokości odbicia. Sam cykl odbijania powtarzać się będzie aż do momentu utraty przez piłkę energii, każde zaś kolejne odbicie będzie powodowało, że piłka odbije się na coraz to mniejszą wysokość. Wiadomo także, że ciężka i miękka piłka odbijać się będzie znacznie słabiej od lekkiej i sprężystej, ale etapy przyspieszania, zwalniania i ponownego przyspieszania będą dla każdej piłki jednakowe. Zależność między rytmem a odstępem w analizowanym przypadku przedstawiono na rysunku 1.1.



Rysunek 1.1: Rysunek przedstawiający animację odbijającej się piłki od podłoża jako przedstawienie zależności między wycuciem czasu oraz odstępem.

Relacje między tymi parametrami można łączyć na wiele sposobów, by w ten sposób animować wszelkiego rodzaju nie tylko fizycznie obiekty.

Wskazówki dla współczesnych animatorów, którzy pracują nie tylko przy filmach, ale i grach czy też robotyce, można znaleźć w „12 zasadach animacji” opisanych w książce „The Illusion of Life: Disney Animation”[14] autorstwa czołowych animatorów studia Disneya, którzy przyczynili się do wytworzenia bardzo charakterystycznego dla tego studia stylu animacji. Spośród dwunastu zasad, siedem można bezpośrednio odnieść do omawianych pojęć wycucia czasu oraz odstępem. Są nimi:

- **Zgniatanie i rozciąganie** (ang. *squash and stretch*).

Zasada ta jest stosowana w celu przekazania dynamiki ruchu i zjawisk związanych z odkształceniem pod wpływem działania sił oraz uznawana za najważniejszą z zasad. Po zastosowaniu nadaje animowanym postaciom i obiektom iluzję grawitacji, wagi, masy i elastyczności. Stosowana łącznie z zasadą „wyolbrzymienia”, może zostać użyta zarówno do prostych obiektów, jak np. odbijającej się piłki, ale także do skomplikowanych struktur, np. ludzkiej twarzy. W przypadku animacji zmierzającej do realizmu konieczne jest

zachowanie stałej objętości obiektu podczas symulowania odkształceń. Mocne przewartościowanie tego środka prowadzi do efektu komizmu.

- **Przewidywanie** (ang. *anticipation*).

Pomaga przygotować widza na to, co ma się wydarzyć. Każde działanie poprzedzone jest przygotowaniem do niego – np. postać, aby podskoczyć musi najpierw zgiąć kolana. Po zastosowaniu wpływa na bardziej realistyczne działanie obiektu. Zasadę można świadomie pominąć, aby uzyskać komediowy efekt zaskoczenia.

- **Nakładanie się i podążanie za akcją** (ang. *follow through and overlapping action*).

Te dwie ściśle związane ze sobą zasady pozwalają utrzymać wrażenie ruchu zgodnego z prawami fizyki. Podążanie za akcją to bezwładne przemieszczanie części postaci (np. włosów) lub obiektu w kierunku działania siły, mimo zakończenia procesu przemieszczania się. Nakładanie się akcji to utrzymanie odpowiedniej relacji czasowej pomiędzy ruchem różnych elementów (na przykład głowy, rąk i nóg w biegu). Podobnie jak w przewidywaniu, mocne przewartościowanie prowadzi do komicznego rezultatu.

- **Rozpędzenie i zwolnienie** (ang. *slow in and slow out*).

Większość obiektów potrzebuje czasu by rozpocząć ruch lub zatrzymać się – trwa to tym dłużej, im większą masę, a więc i bezwładność, ma dane ciało. Animacja jest bardziej realistyczna, jeśli zawiera więcej klatek przy początku i końcu ruchu niż w środku.

- **Łuki** (ang. *arc*).

W procesie animacji powinno się uwzględniać to, że kiedy obiekty się poruszają, to podążają w polu grawitacyjnym trajektoriami, które są łukami. Animacje zatem powinny odzwierciedlać ten łuk, by być zgodnymi z prawami fizyki.

- **Taktowanie** (ang. *timing*).

Korzystanie z właściwego „czasu” pozwala kontrolować nastrój i reakcję postaci lub obiektów. Czas ruchu musi odnosić się zarówno do praw fizyki, jak i do historii opowiedzianej przez animację. Lekka przesada może być tu użyta, ale z wyczuciem i w sposób konsekwentny.

- **Wyolbrzymienie** (ang. *exaggeration*).

Animacja nie musi być całkowicie realistyczna, odrobina przesady jest wręcz wskazana i dodaje animacji dynamizmu.

1.3 Metoda klatek kluczowych w animacji komputerowej

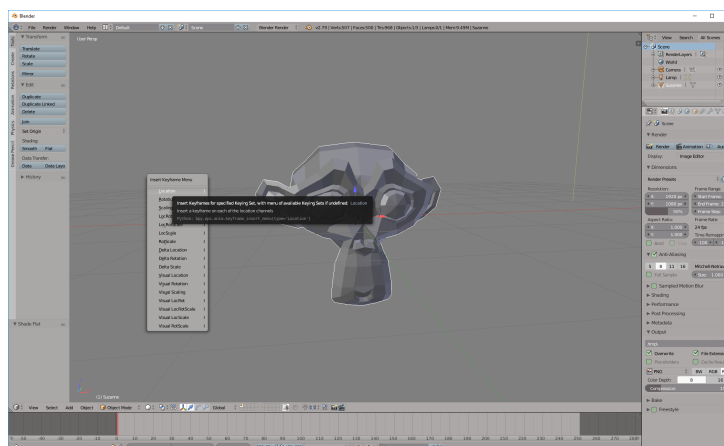
Trójwymiarowa animacja komputerowa wykorzystywana jest w przemyśle filmowym, w produkcjach multimedialnych oraz w aplikacjach czasu rzeczywistego, nie tylko gier wideo ale także specjalistycznych symulatorów szkoleniowych (służb mundurowych oraz wojska). Ten

rodzaj animacji, trójwymiarowa animacja komputerowa, jest głównym obiektem zainteresowania w prezentowanej pracy.

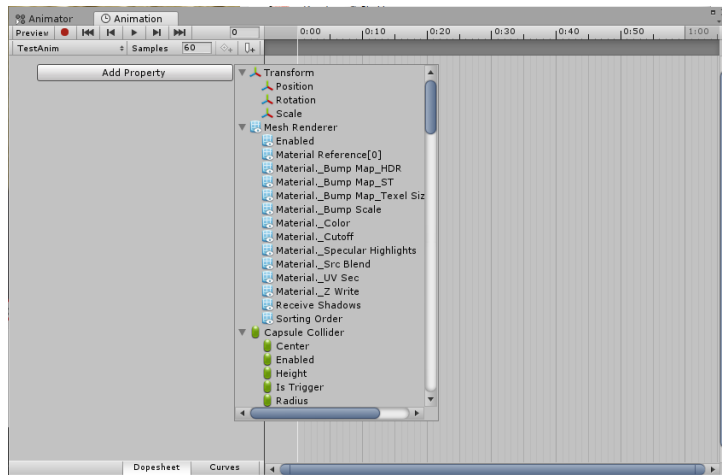
Metoda klatek kluczowych jest jedną z możliwych metod animacji (dostępne metody w animacji komputerowej opisane zostały w dodatku C), która to metoda ma korzenie w tradycyjnej animacji odręcznej. Polega ona na wyznaczaniu przez animatora wartości parametrów dla podawanych animacji klatek kluczowych, natomiast wartości pośrednie, między klatkami kluczowymi, obliczane są przez algorytm komputerowy. W tej metodzie artysta ma kluczową rolę oraz bardzo dużą kontrolę nad powstawaniem animacji. Metoda klatek kluczowych jest powszechnie używana we wszystkich najważniejszych aplikacjach do produkcji filmów (animowanych czy efektów specjalnych) oraz we wszystkich silnikach gier wideo i symulatorów.

W animacji komputerowej, w odróżnieniu od animacji tradycyjnej, pojęcie metody klatek kluczowych może być nieco mylące. Powinno się raczej mówić o metodzie parametrów kluczowych, gdyż artysta animuje tylko wybrane przez niego, konieczne w animacji, parametry dowolnego obiektu występującego na scenie. Artysta nie pracuje nad ostatecznym wyglądem całej klatki obrazu, jak ma to miejsce w metodzie tradycyjnej (np. w procesie powstawania filmów animowanych Walta Disneya).

Parametrami, które mogą być animowane, są nie tylko właściwości w przestrzeni trójwymiarowej obiektów, takie jak współrzędne, kąty obrotu względem osi, skala obiektu, ale także właściwości materiału, z którego jest zbudowany obiekt, takie jak: kolor, stopień przezroczystości, chropowatość, połysk itp. Nic nie stoi na przeszkodzie, aby animować parametry samej wirtualnej kamery, takie jak jej kąt oraz zasięg widzenia. Każdy parametr, który zostanie udostępniony animatorowi, czy to poprzez zmienną w języku animacji [22] czy to w graficznym edytorze, jako element interfejsu (co pokazane jest na rysunkach 1.2 i 1.3), może być wykorzystywany w tej metodzie.



Rysunek 1.2: Rysunek przedstawiający dostępne parametry w programie graficznym Blender.

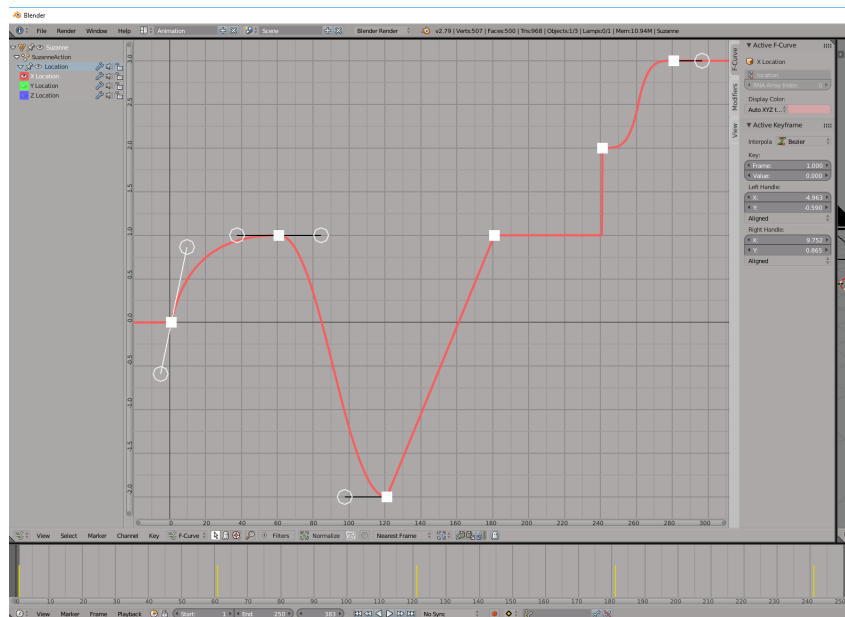


Rysunek 1.3: Rysunek przedstawiający dostępne parametry w silniku gier wideo Unity.

Każdemu parametrowi w klatce kluczowej należy przypisać pewną wartość, która będzie decydowała o zachowaniu się tego parametru w trakcie animacji.

Klatka kluczowa to para uporządkowana dwóch wartości (x, y) (zwana także kluczem (ang. *key-frame*)), gdzie zmienna x reprezentuje numer klatki o wartości y animowanego parametru wybranego przez animatora. Sekwencje kluczy dla jednego parametru nazywane są w tej metodzie ścieżkami (ang. *tracks*) lub kanałami (ang. *channels*).

Pośrednie wartości parametrów między kluczami następnie są obliczane według z góry ustalonej procedury. W animacji komputerowej te procedury są implementowane w aplikacjach przez różnego rodzaju krzywe interpolacyjne do sterowania zmian wartości parametrów w czasie pomiędzy klatkami kluczowymi (co zostało przedstawione na rysunku 1.4).



Rysunek 1.4: Graficzny edytor programu Blender przedstawiający klatki kluczowe (punkty koloru pomarańczowego), oraz wykresy różnego rodzaju funkcji interpolacyjnych (krzywe koloru czerwonego). Dla interpolacji krzywymi Béziera okręgi (koloru białego) odpowiadają uchwytem.

1.4 Właściwości krzywych interpolacyjnych

Interpolacja w metodzie klatek kluczowych jest to metoda obliczania wartości pośrednich animowanego parametru pomiędzy klatkami kluczowymi. Nachylenie krzywej określa szybkość, z jaką wartość parametru zmienia się między klatkami kluczowymi. Strome nachylenie krzywej oznacza, że dana wartość zmienia się szybko, łagodne oznacza zaś, że wartość ta zmienia się wolno. Przegląd najważniejszych rodzajów funkcji interpolacyjnych zamieszczono w dodatku D.

W metodzie klatek kluczowych nie ma żadnej przeszkody dla wykorzystania w interpolacji pomiędzy kluczami dowolnej funkcji. Ważne jest, by ta funkcja była wyznaczona jednoznacznie (dla wybranego argumentu zawsze przyjmowała tę samą wartość) oraz żeby możliwym było przedstawienie jej w formie matematycznego wzoru.

Aby wybrać tę najbardziej odpowiednią funkcję, w konkretnym przypadku, należy odpowiedzieć na następujące pytania:

- **Jaka jest potrzebna postać wzoru krzywej interpolacyjnej?**

Wzór krzywej interpolacyjnej może być zdefiniowany na trzy sposoby: jawny, niejawny (uwikłany) oraz parametryczny.

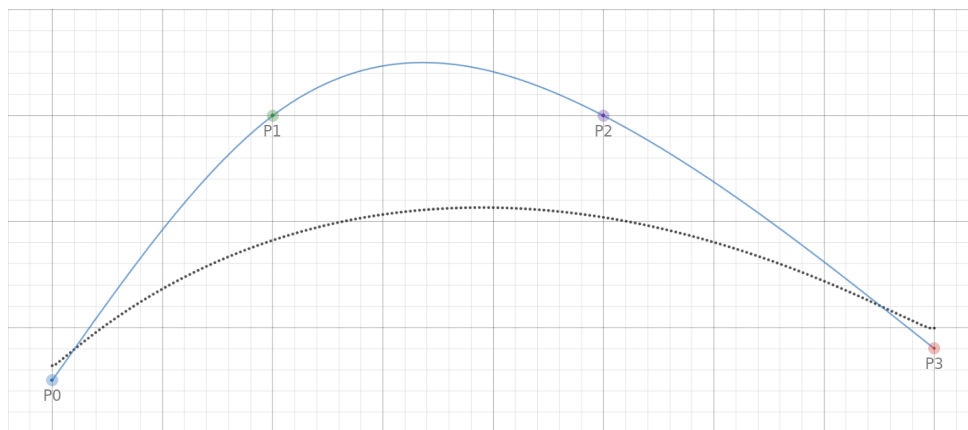
Sposób jawny to przedstawienie wzoru krzywej np. $y = f(x)$, ten sposób bezpośrednio pozwala wyznaczać wartość parametru y dla dowolnego argumentu x .

Sposób niejawnny polega na podaniu równania w postaci $f(x, y) = 0$. W tym przypadku rozpatrywaną krzywą wyznacza poziomica zerowa funkcji f . Dysponując tym równaniem możliwym jest sprawdzenie, czy dany punkt leży na krzywej, badając, czy jego współrzędne x oraz y spełniają to równanie.

Sposób parametryczny polega na użyciu wzorów: $x = f(t)$, $y = g(t)$ gdzie $f, g : [t_{min}, t_{max}] \rightarrow \mathbb{R}$ są funkcjami, a $t \in [t_{min}, t_{max}]$. To podejście pozwala na łatwą konstrukcję krzywych w przestrzeni \mathbb{R}^2 , gdyż wystarczy podstawić dowolny argument z dziedziny do wyznaczenia punktu o współrzędnych (x, y) .

- **Interpolacja czy aproksymacja?**

Czy krzywa powinna przechodzić przez wyznaczone wartości parametru w klatce kluczowej? Jeżeli tak, to mówimy tutaj o interpolacji i szukamy pomocniczych krzywych, gdzie najprostszym rozwiązaniem jest interpolacja liniowa. Jeżeli nie, jest to problem aproksymacyjny i te miejsca, przez które krzywa nie musi przechodzić, stają się punktami kontrolnymi, które pomagają wyznaczyć ostateczny kształt krzywej (jak to ma miejsce w konstrukcji krzywej Béziera). Obie te właściwości przedstawione zostały na rysunku 1.5.



Rysunek 1.5: Wykres funkcji interpolacyjnej (ciągła niebieska krzywa), oraz wykres funkcji aproksymującej (czarna przerywana krzywa).

- **Jaka jest złożoność obliczeniowa funkcji interpolacyjnej?**

Złożoność obliczeniowa to szczególnie ważny aspekt animacji komputerowych czasu rzeczywistego, gdyż przekłada się bezpośrednio na czas jakiego potrzebuje procesor na wyznaczenie wartości pośrednich parametrów. Im prostszy wzór funkcji, tym szybciej przebiegają obliczenia, co przekłada się na większą liczbę możliwych parametrów, jakie mogą być animowane w jednej jednostce czasu. Dla gier wideo jest to o tyle istotne, że pozwala animować jednocześnie większą liczbę obiektów, w tym postaci ludzkich.

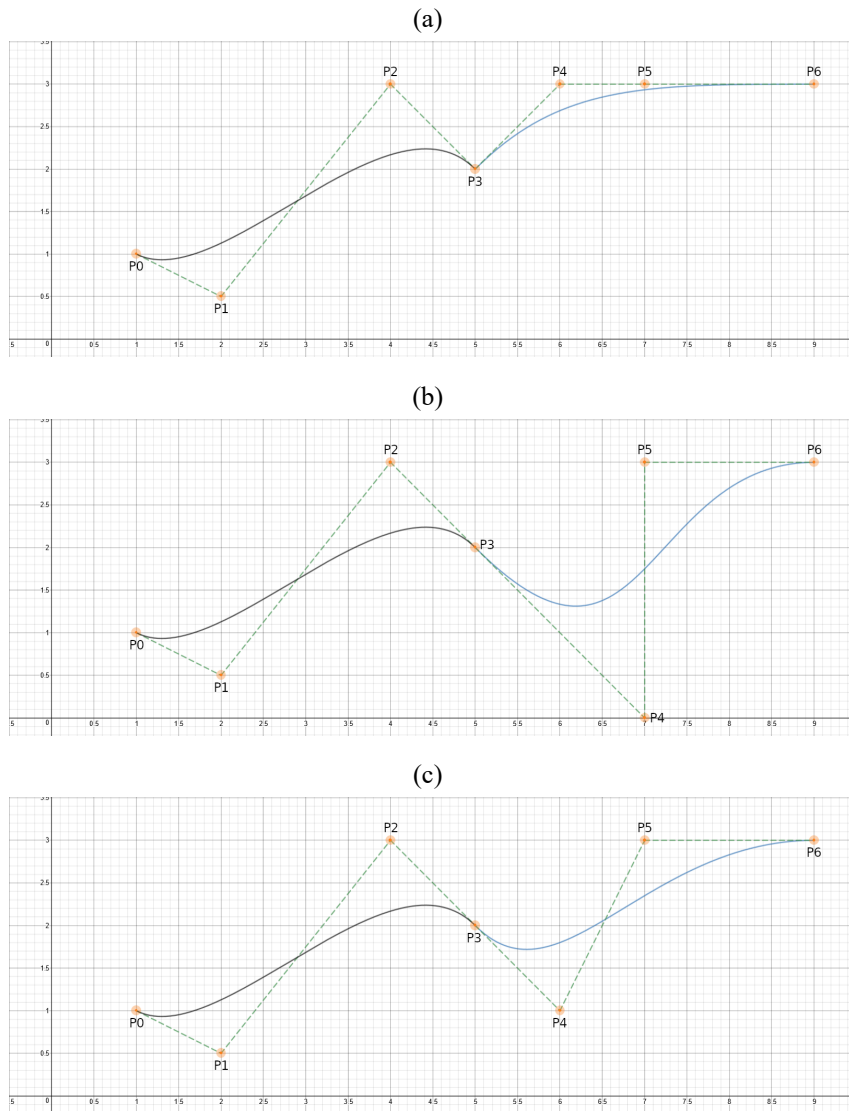
- **Jaka jest klasa gładkości krzywej?**

Pojęcie ciągłości funkcji f (funkcji klasy C^0) na odcinku (a, b) definiowane jest przy

pomocy pojęcia ciągłości w punkcie: f jest ciągła w punkcie $x_0 \in (a, b)$ jeśli $\lim_{x \rightarrow x_0} f(x) = f(x_0)$, $x \in (a, b)$ i $x \neq x_0$; jeśli f jest ciągła w każdym punkcie x odcinka (a, b) wówczas mówimy, że f jest ciągła na odcinku (a, b) , co zapisujemy $f \in C^0(a, b)$. Jeśli ponadto istnieje pochodna funkcji f w każdym punkcie (a, b) i pochodna ta jest funkcją ciągłą w każdym punkcie odcinka (a, b) wtedy mówimy, że f jest klasy C^1 na (a, b) (i zapisujemy $f \in C^1(a, b)$). Ogólnie funkcja f jest klasy C^m na (a, b) $m = 0, 1, \dots$, jeśli f ma pochodną rzędu m , ciągłą na (a, b) . Liczbę m nazywamy klasą gładkości funkcji f .

Innym zagadnieniem jest klasa gładkości funkcji w punktach końcowych odcinka (węzłach), co przedstawione jest na rysunku 1.6. Jeżeli funkcja f jest określona na (a, b) , funkcja g jest określona na (b, c) , $a < b < c$, a ponadto f ma granicę lewostronną w punkcie b , natomiast g ma granicę prawostronną w punkcie b , to w przypadku, gdy granice te są równe powiemy, że te „sąsiadujące” funkcje można skleić w klasie C^0 (uzyskamy przedłużenie na odcinek (a, c) do funkcji ciągłej). W dalszym ciągu taką parę funkcji f oraz g będzie nazywana **funkcjami sąsiadującymi**, przedłużalnymi w klasie C^0 . W przypadku gdy funkcje sąsiadujące są klasy C^m i tak uzyskane przedłużenie na odcinek (a, c) jest funkcją klasy C^m mówimy że para f, g jest przedłużalna (inaczej: sklejalna w punkcie b) w klasie C^m .

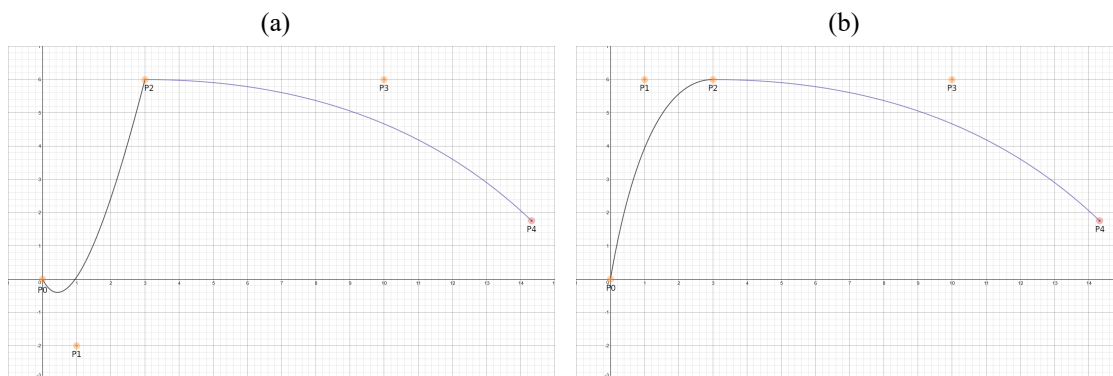
Oczywiście zachowanie ciągłości funkcji opisujących ruch nie jest konieczne, o ile taki jest zamiar animatora.



Rysunek 1.6: Przedstawienie wykresów sąsiadujących funkcji o ciągłości C^0 oraz gładkości klasy C^1, C^2 , w punkcie wspólnym. (a) ciągłość C^0 , (b) gładkość klasy C^1 , (c) gładkość klasy C^2 .

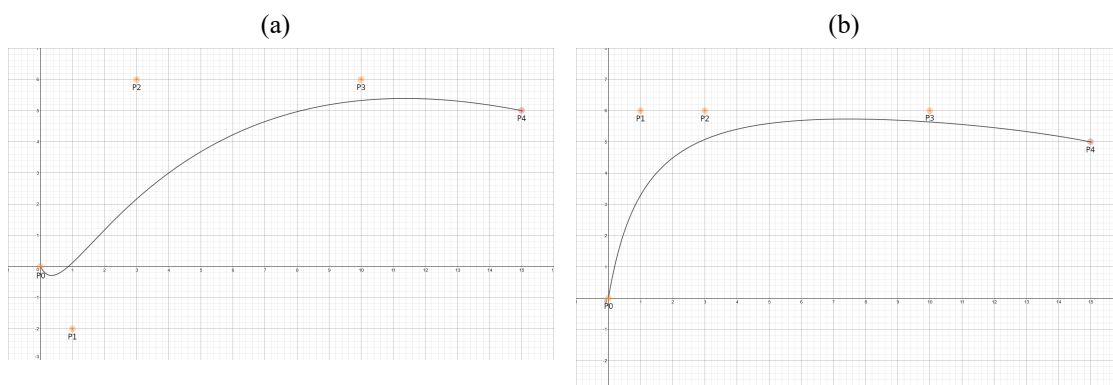
- **Jaki wybrać sposób kontroli kształtu?**

Możliwe są dwa sposoby kontroli kształtu krzywej. Lokalna kontrola kształtu krzywej objawia się tym, że zmiana jednego punktu kontrolnego ma wpływ na ograniczoną część krzywej jedynie w jego okolicy (np. interpolacja krzywymi Catmulla-Roma) - rysunek 1.7.



Rysunek 1.7: Przedstawienie lokalnej kontroli kształtu. Zmiana pozycji punktu kontrolnego P1 z pozycji jak na wykresie z lewej strony na pozycje jak na wykresie z prawej wpływa na kształt tylko krzywej koloru czarnego. (a) Wykres krzywej przed zmianą pozycji, (b) wykres krzywej przed zmianą pozycji punktu kontrolnego P1, (b) wykres krzywej po zmianie pozycji punktu kontrolnego P1.

Przeciwieństwem lokalnej kontroli kształtu jest globalna kontrola kształtu, gdzie zmiana jednego punktu kontrolnego ma wpływ na krzywą na całej jej długości (np. w interpolacji krzywymi Béziera wyższych stopni), co pokazuje rysunek 1.8



Rysunek 1.8: Przedstawienie globalnej kontroli kształtu. Zmiana pozycji punktu kontrolnego P1 z pozycji jak na wykresie z lewej strony na pozycje jak na wykresie z prawej wpływa na kształt krzywej na całej jej długości. (a) Wykres krzywej przed zmianą pozycji punktu kontrolnego P1, (b) wykres krzywej po zmianie pozycji punktu kontrolnego P1.

Balansowanie między powyższymi właściwościami, przy wyborze odpowiedniego rodzaju krzywej interpolacyjnej, nie jest łatwym zadaniem. Niemniej jednak współczesne edytory animacji, wbudowane w programy graficzne, takie jak Blender, 3ds Max czy też Maya, udostępniają tylko skończoną liczbę rodzajów krzywych, które sprawdzają się w praktyce od szeregu lat. Dokładniejszy ich opis zostanie przedstawiony w dodatku D.

1.5 Reprezentacja krzywych interpolacyjnych w postaci jednej krzywej Béziera trzeciego stopnia

Stosowane w animacji krzywe interpolacyjne, opisane w dodatku D, w większości są możliwe do reprezentowania w postaci krzywej Béziera trzeciego stopnia (o punktach kontrolnych P'_0, P'_1, P'_2, P'_3), dokładnie albo z określonym błędem aproksymacji.

- **Funkcja schodkowa**

Funkcji schodkowej nie można przedstawić w postaci jednej krzywej Béziera trzeciego stopnia, z powodu braku zachowania ciągłości C^0 w klatkach kluczowych.

- **Funkcja liniowa**

Funkcja liniowa interpolująca wartości w klatkach kluczowych P_0 oraz P_1 jest możliwa do zapisu w postaci krzywej Béziera trzeciego stopnia, gdzie punkty kontrolne krzywej Béziera można wyznaczyć za pomocą wzorów:

$$\begin{cases} P'_0 = P_0 \\ P'_1 = \frac{1}{3}(P_0 + P_1) \\ P'_2 = \frac{2}{3}(P_0 + P_1) \\ P'_3 = P_1 \end{cases} \quad (1.1)$$

- **Funkcja wygładzająca**

W pracy [13] wykazano, że każda funkcja wygładzająca (ang. *easing functions*), w literaturze znane także jako funkcje Pennera [27] (z określonym małym błędem aproksymacji według miary błędu średnio-kwadratowego) może być wyrażona w postaci krzywej Béziera trzeciego stopnia. Więcej na temat funkcji wygładzających można znaleźć w dodatku D.

- **Krzywa Hermite'a trzeciego stopnia**

Wzór na interpolację krzywej Hermite'a trzeciego stopnia można przedstawić jako:

$$H(t) = (2t^3 - 3t^2 + 1)P_0 + (t^3 - 2t^2 + t)M_0 + (-2t^3 + 3t^2)P_1 + (t^3 - t^2)M_1 \quad t \in [0, 1] \quad (1.2)$$

gdzie P_0 to punkt startowy wraz jego z styczną M_0 , a P_1 to punkt końcowy z jego styczną M_1 . Reprezentacja Béziera krzywej danej w tej postaci składa się z punktów kontrolnych P'_0, P'_1, P'_2, P'_3 , które można obliczyć na podstawie wzoru (1.3).

$$\begin{cases} P'_0 = P_0 \\ P'_1 = P_0 + \frac{1}{3}M_0 \\ P'_2 = P_1 - \frac{1}{3}M_1 \\ P'_3 = P_1 \end{cases} \quad (1.3)$$

- **Krzywa Catmulla-Roma**

Korzystając z pracy [44] oraz zakładając, że P_0, P_1, P_2, P_3 są kolejnymi punktami kontrolnymi krzywej Catmulla-Roma, otrzymujemy wzory na punkty kontrolne krzywej Béziera trzeciego stopnia:

$$\left\{ \begin{array}{l} P'_0 = P_1 \\ P'_1 = d_1^{2\alpha} P_2 - d_2^{2\alpha} P_0 + (2d_1^{2\alpha} + 3d_1^\alpha d_2^\alpha + d_2^{2\alpha}) P_1 \\ P'_2 = d_3^{2\alpha} P_2 - d_2^{2\alpha} P_3 + (2d_3^{2\alpha} + 3d_3^\alpha d_2^\alpha + d_2^{2\alpha}) P_2 \\ P'_3 = P_2 \\ d_1 = |P_1 - P_0| \\ d_2 = |P_2 - P_1| \\ d_3 = |P_3 - P_2| \end{array} \right. \quad (1.4)$$

gdzie α to współczynnik określający dodatkową parametryzację opisaną wyżej: gdy $\alpha = 0$ krzywa występuje w parametryzacji dośrodkowej (ang. *centripetal*), gdy $\alpha = 0.5$ - w parametryzacji jednostajnej (ang. *uniform*), zaś gdy $\alpha = 1$ - w parametryzacji łukowej (ang. *chordal*). Więcej na temat opisanych parametryzacji można znaleźć w pracy [44].

- **Krzywa TCB (Kochanka-Bartelsa)**

Wzór na krzywą TCB [17] jest tożsamy z wzorem dla krzywej Hermite'a trzeciego stopnia [59], z tą różnicą, że styczne M_0 oraz M_1 są zależne od parametrów zwanych napięciem, gładkością oraz skosem i wyznaczone są przez wzory:

$$\left\{ \begin{array}{l} M_0 = \frac{1}{2}(1-t)(1+b)(1+c)(P_0 - P'_0) + \frac{1}{2}(1-t)(1-b)(1-c)(P_1 - P_0) \\ M_1 = \frac{1}{2}(1-t)(1+b)(1-c)(P_1 - P_0) + \frac{1}{2}(1-t)(1-b)(1+c)(P'_1 - P_1) \end{array} \right. \quad (1.5)$$

gdzie t, c, b to parametry kolejno napięcia, gładkości oraz skosu, a P'_0 to punkt startowy sąsiadującej krzywej przed aktualną krzywą, a P'_1 to punkt końcowy sąsiadującej krzywej aktualnej krzywej.

Dzięki przedstawionym przekształceniom najbardziej popularne typy interpolacji (oprócz funkcji schodkowej) można zapisać i interpolować tylko przy użyciu interpolacji jedną krzywą Béziera trzeciego stopnia.

Więcej na temat krzywych interpolacyjnych w dodatku D.

1.6 Funkcje interpolacyjne w praktyce

W zależności od tego, czy interpolacje parametrów wykonywane są w aplikacjach czasu rzeczywistego (takich jak gry wideo czy symulatory cywilne lub wojskowe) czy też w edyto-

rach graficznych generujących filmy w trybie offline, algorytmy oraz podejście do problemów wydajnościowo-pamięciowych wyglądają w praktyce inaczej.

1.6.1 Aplikacje generujące obrazy komputerowe pracujące w trybie offline

Proces tworzenia filmu animowanego trwa znacznie dłużej, niż wyświetlanie samego filmu. Co więcej, twórcy mogą sobie pozwolić na dowolne wydłużanie czasu produkcji, aby uzyskać zamierzony efekt fotorealistyczny lub stylistyczny.

Na czas generowania jednej, fotorealistycznej lub stylizowanej, klatki filmu animowanego składa się nie tylko konieczność wyznaczenia pozycji, orientacji oraz pozostałych parametrów animowanych obiektów widocznych na scenie, ale przede wszystkim obliczenie oświetlenia danej sceny. W tym celu wykorzystuje się, w zależności od programu graficznego, różnego rodzaju algorytmy globalnego oświetlenia (ang. *global illumination*) [7], które to algorytmy obliczają oświetlenie trójwymiarowej sceny nie tylko od promieni światła pochodzącego bezpośrednio od jego źródła (tak zwane oświetlenie bezpośrednie), ale także od tych promieni świetlnych, które odbijane są od innych powierzchni występujących na scenie (oświetlenie pośrednie). Teoretycznie odbite lub załamane promienie świetlne oraz cienie są przykładami globalnego oświetlenia, ponieważ symulując je, jeden obiekt wpływa na kreowanie obrazu innego obiektu. W praktyce jednak tylko symulacja rozproszonego odbicia lub wyznaczanie obrazu kaustyki nazywane są globalnym oświetleniem.

Każdy z programów graficznych, takich jak 3d Max, Maya, Blender, zawierają moduł (jeden albo kilka) odpowiedzialny za kreowanie obrazów fotorealistycznych scen, który to moduł implementuje jeden z algorytmów globalnego oświetlenia. I tak np. program Blender implementuje jednokierunkowe śledzenie ścieżek z wielokrotnym próbkowaniem ważonym (ang. *Unidirectional path tracing with multiple importance sampling*) o nazwie własnej Cycles [55]. Wykreowanie fotorealistycznego (stylizowanego) obrazu trójwymiarowej sceny to problem o wiele bardziej skomplikowany, niż wykorzystanie algorytmów odpowiedzialnych za animacje obiektów na scenie. Do wykreowania obrazu przykładowej sceny, w rozdzielczości HDTV (1920x1080 pikseli), liczba promieni koniecznych do oświetlenia sceny przekracza 10^7 , gdyż sama liczba promieni pierwotnych wynosi około 2×10^6 . Z drugiej strony liczba interpolowanych parametrów wszystkich animowanych obiektów na scenie, która może sięgać poziomem 10^5 (dla przykładowej sceny: 300 postaci \times 60 kości jednej postaci \times 6 interpolowanych parametrów dla kości).

Na przykładzie programu Blender, wolnego i otwartego oprogramowania do modelowania i kreowania obrazów oraz animacji trójwymiarowych, pokazać można także, jak wyglądają algorytmy odpowiedzialne za animację metodą klatek kluczowych. Algorytm obliczania wartości

dla dowolnego interpolowanego parametru można przedstawić w postaci listingu (oryginalny kod algorytmu można znaleźć pod adresem repozytorium [60]).

Algorytm 1 Listing algorytmu interpolacji parametrów w metodzie klatek kluczowych w programie Blender.

```
function INTERPOLACJA PARAMETRU(rodzaj interpolacji, czas, parametry)
  if rodzaj interpolacji = funkcja wygładzająca then
    wywołaj funkcje wygładzające(czas, parametry)
  else if rodzaj interpolacji = krzywa Bézier then
    wywołaj funkcje Bézier(czas, parametry)
  else if rodzaj interpolacji = funkcja liniowa then
    wywołaj funkcje liniowa(czas, parametry)
  else if rodzaj interpolacji = funkcja skokowa then
    wywołaj funkcje skokowa(czas, parametry)
  end if
end function
```

Szczególną uwagę należy zwrócić na interpolację przy użyciu krzywej Béziera (wywołaj funkcje Bézier) z listingu Algorytm 2,

Algorytm 2 Listing algorytmu interpolacji przy użyciu krzywej Béziera trzeciego stopnia.

```
function WYWOŁAJ FUNKCJE BÉZIER( $t, p_{0x}, p_{1x}, p_{2x}, p_{3x}, p_{0y}, p_{1y}, p_{2y}, p_{3y}$ )
   $s = \text{znajdź\_pierwiastek}(t, p_{0x}, p_{1x}, p_{2x}, p_{3x})$ 
   $d = p_{0y}$ 
   $c = 3(p_{2y} - p_{0y})$ 
   $b = 3(p_{0y} - p_{1y} + p_{2y})$ 
   $a = p_{3y} - p_{0y} + 3(p_{1y} - p_{2y})$ 
   $val = d + cs + bs^2 + as^3$ 
  return  $val$ 
end function
```

gdzie funkcja `znajdź_pierwiastek` odpowiedzialna jest za wyznaczenie przeciwobrazu zadanej wartości zmiennej t dla wielomianu co najwyżej trzeciego stopnia. Ten algorytm korzysta z wzorów Cardano i wydaje się trywialny. Jednak analizując przykład implementacji (Algorytm 3) można zauważyć znaczne obciążenie obliczeniowe, które może być istotne przy obliczeniach animacji, realizowanych nawet na współczesnych procesorach.

Algorytm 3 Listing metody do wyznaczania miejsca zerowego wielomianu co najwyżej trzeciego stopnia.

```
function ZNAJDŹ_PIERWIASTEK( $t, p0_x, p1_x, p2_x, p3_x$ )
   $c0 = p0_x - t$ 
   $c1 = 3 * (p1_x - p0_x)$ 
   $c2 = 3 * (p0_x - 2 * p1_x + p2_x)$ 
   $c3 = p3_x - p0_x + 3 * (p1_x - p2_x)$ 
  if  $c3 \neq 0$  then
     $a = c2/c3$ 
     $b = c1/c3$ 
     $c = c0/c3$ 
     $a = a/3$ 
     $p = b/3 - a^2$ 
     $q = (2 * a^3 - a * b + c)/2$ 
     $d = q^2 + p^3$ 
     $out = \text{znajdź pierwiastek funkcji sześcienniej}(t, a, b, c, d)$ 
  else
     $a = c2$ 
     $b = c1$ 
     $c = c0$ 
    if  $a \neq 0$  then
       $out = \text{znajdź pierwiastek funkcji kwadratowej}(t, a, b, c)$ 
    else if  $b \neq 0$  then
       $out = \text{znajdź pierwiastek funkcji liniowej}(t, b, c)$ 
    else if  $c \neq 0$  then
       $out = 0$ 
    end if
  end if
  return  $out$ 
end function
```

Postępowanie takie, przedstawione w listingu 2 lub w listingu 3, pozwala interpolować parametry w metodzie klatek kluczowych przy użyciu krzywej Béziera w parametryzacji (3.4), opisanej szerzej w rozdziale 3.1.

Listing 1 wyznaczania parametrów, z uwzględnieniem czterech możliwych sposobów interpolacji parametrów w metodzie klatek kluczowych, w znikomym stopniu wpływa na czas tworzenia jednej klatki filmu animowanego. Twórcy programów graficznych pracujących w trybie offline nie przykładają dużej wagi do wydajności obliczeniowej algorytmu wyznaczania wartości funkcji interpolacyjnych w metodzie klatek kluczowych, natomiast bardziej zwracają uwagę na intuicyjny interfejs oraz liczbę funkcji oddanych w ręce artystom. Rozwijane oraz optymalizowane są tylko algorytmy odpowiedzialne ze globalne oświetlenie, a także techniki definiowana (budowania) coraz to bardziej fotorealistycznych materiałów oraz udostępniane są użytkownikom coraz to nowsze funkcje wspomagające oraz przyspieszające produkcję animacji.

1.6.2 Aplikacje czasu rzeczywistego

Współczesne gry wideo to dynamiczne oraz interaktywne symulacje czasu rzeczywistego, gdzie czas odgrywa bardzo ważną rolę [10]. I tak dla programistów silników gier wideo najważniejszym celem jest, aby czas generowania jednej klatki animacji był jak najmniejszy, aby w ten sposób uzyskiwać jak największą liczbę generowanych klatek na sekundę (ang. *frames per second*). Właśnie tę miarę, liczbę klatek na sekundę, programiści silników graficznych traktują jako prawdziwą miarę wydajności ich programów. Liczba klatek na sekundę to nic innego jak częstotliwość klatek wyrażona w Hz, czyli 1/s - odwrotność czasu trwania jednej klatki. Z tego względu jest to miara nieliniowa, nie należy porównywać bezwzględnych wartości liczby klatek na sekundę. Jest jednak dobra do określenia, czy generowana animacja w wirtualnym świecie jest postrzegana przez odbiorców jako płynna czy skokowa. Subiektywny efekt płynności można uzyskać już przy 24 – 30Hz, ale zwiększenie tej wartości do 60Hz a nawet do 144Hz odbierana jest przez użytkowników jako lepszą, bardziej płynną.

Programiści gier wideo narzucają sobie przy budowaniu gry górne ograniczenie czasu trwania jednej klatki (dla współczesnych gier ta wartość wynosi 16.6ms lub 33.3ms odpowiednio 60Hz lub 30Hz [25]). Wartość ta jest równa częstotliwości (lub jej wielokrotności) odświeżania większości obecnych monitorów lub telewizorów. Przy okazji wartość ta jest rozsądnym kompromisem pomiędzy czasem potrzebnym na symulowanie złożonych wirtualnych światów oraz coraz to bardziej fotorealistycznej wizualizacji, a zachowaniu dużej liczby klatek na sekundę. Dodatkowo ustalenie jednego stałego czasu jednej klatki symulacji pozwala na synchronizację czynności wykonywanych przez wszystkich graczy w grach wieloosobowych.

Mimo że częstotliwość 60Hz jest bardzo często spotykana w monitorach oraz telewizorach obecnych na rynku, w 2014 roku organizacja VESA ogłosiła dodanie nowego standardu dla interfejsu DisplayPort 1.2 o nazwie „Synchronizacji Adaptacyjnej” (ang. *Adaptive-Sync*) [56], Technologia stojąca za tym standardem umożliwia sterowanie, przez aplikację, częstotliwością odświeżania monitora komputerowego, w pewnych ustalonych zakresach częstotliwości.

Na czas jednej klatki symulacji składają się czasy wykonywania podprogramów takich jak symulacja praw fizycznych, tworzenie realistycznej grafiki, obliczenia związane z animacją oraz zachowaniami aktorów na scenie, obsługa urządzeń wejścia [10]. Suma wykonywania wszystkich podprogramów klatki musi być mniejsza niż ustalony czas jednej klatki, gdyż przekroczenie tego czasu powoduje, że klatka symulacji trwa dwa razy dłużej, a to odbija się niekorzystnie dla liczby klatek na sekundę. Dlatego też bardzo duży nacisk kładziony jest na optymalizację, a co za tym idzie, szybkość działania każdego z podprogramów.

Marginalny dla aplikacji pracujących w trybie offline problem wydajnościowy w interpolowaniu parametrów w metodzie klatek kluczowych, w aplikacjach czasu rzeczywistego jest już nie do pominięcia (gdy potrzeba jest ożywić coraz to większą liczbę obiektów na wirtualnej scenie). Dlatego też w podprogramie odpowiedzialnym za obliczenia związane z animacjami

obiektów na scenie preferowana jest okrojona różnorodność w liczbie dostępnych funkcji (opisanych w dodatku A), kosztem zwiększenia liczby wykonywanych operacji (co za tym idzie możliwość animowania większej liczby parametrów, obiektów, postaci) w ciągu jednej klatki generowanej sceny.

Do produkcji animacji na potrzeby gier wideo powszechnie wykorzystuje się te same narzędzia, co do produkcji filmów animowanych. Programy te mają na tyle rozbudowane narzędzia do modelowania obiektów trójwymiarowych, animacji, teksturowania, projektowaniu materiałów itp., że stały się użyteczne w obu zastosowaniach. Formaty do zapisu animacji (takie jak FBX[®] [61], COLLADA[™][47], OpenGEX [48]), które są potrzebne jako pliki wymiany pomiędzy aplikacjami graficznymi (lub pomiędzy aplikacjami do generowania modeli trójwymiarowych oraz animacji a silnikami gier wideo), powszechnie obsługują tylko trzy funkcje interpolacyjne dla metody klatek kluczowych. Są nimi: funkcja schodkowa, funkcja liniowa i krzywa Béziera trzeciego stopnia.

Propozycje implementacji funkcji interpolacyjnych w silnikach aplikacji czasu rzeczywistego można znaleźć w dokumencie formatu plików OpenGEX [48] i przedstawia się następująco: Algorytm interpolacji parametrów jest dwuetapowy. W pierwszym etapie wyznaczany jest parametr s dla danego czasu t .

- Dla funkcji liniowej przedstawia się następująco

$$s(t) = \frac{t - t_1}{t_2 - t_1} \quad (1.6)$$

gdzie t_1 oraz t_2 określone chwile klatek kluczowych.

- Dla krzywej Béziera trzeciego stopnia wyznaczenie s wiąże się z rozwiązaniem równania

$$(1 - s)^3 t_1 + 3s(1 - s)^2 c_1 + 3s^2(1 - s) c_2 + s^3 t_2 - t = 0 \quad (1.7)$$

gdzie t_1, c_1, t_2, c_2 to współrzędne x odpowiednio kolejnych punktów kontrolnych krzywych Béziera trzeciego stopnia. Jeśli t_1, c_1, t_2, c_2 spełniają zależność $t_1 < c_1 < c_2 < t_2$ to równanie (1.7) ma tylko jedno rozwiązanie, które można szybko znaleźć poprzez zastosowanie iteracyjnej metody Newtona, poczynszyszy od $s_0 = s(t)$ wyznaczonym z przypadku liniowego (1.6) i poprawianym przez:

$$s_{i+1} = s_i - \frac{(t_2 - 3c_2 + 3c_1 - t_1)s_i^3 + 3(c_2 - 2c_1 + t_1)s_i^2 + 3(c_1 - t_1)s_i + t_1 - t}{3(t_2 - 3c_2 + 3c_1 - t_1)s_i^2 + 6(c_2 - 2c_1 + t_1)s_i + 3(c_1 - t_1)} \quad (1.8)$$

aż do uzyskania wartości o satysfakcjonującej dokładności.

Po znalezieniu wartości parametru s , w drugim etapie wyznaczana jest wartość parametru v :

- Dla funkcji skokowej

$$v(s) = v_1 \quad (1.9)$$

gdzie v_1 wartość początkowa interpolowanego parametru.

- Dla funkcji liniowej

$$v(s) = (1 - s)v_1 + sv_2, \quad s \in [0, 1] \quad (1.10)$$

gdzie v_1 oraz v_2 to wartości interpolowanego parametru, odpowiednio początkowa oraz końcowa.

- Dla krzywej Béziera trzeciego stopnia

$$v(s) = (1 - s)^3 v_1 + 3s(1 - s)^2 p_1 + 3s^2(1 - s)p_2 s^3 v_2, \quad s \in [0, 1] \quad (1.11)$$

gdzie v_1, p_1, p_2, v_2 to współrzędne y kolejnych punktów kontrolnych krzywych Béziera trzeciego stopnia.

1.7 Wnioski

Analizując przedstawione w poprzednich rozdziałach właściwości interpolacji można byłoby wyciągnąć proste wnioski związane z praktyką stosowania odpowiednich algorytmów. Interpolując parametry w metodzie klatek kluczowych z użyciem krzywej Béziera trzeciego stopnia, płaci się dość wysoką cenę za wygodę w ich konstruowaniu, w postaci rozbudowanego algorytmu. Wygoda w budowaniu rozumiana jest jako możliwość bardzo łatwej i intuicyjnej metody konstruowania animacji ze zmienną prędkością w interpolowaniu parametrów, poprzez zmianę położenia tylko dwóch punktów (uchwyty krzywej Béziera trzeciego stopnia). Algorytm interpolacji parametrów składa się z dwóch etapów i konieczności iteracyjnego znajdowania parametru s , dlatego powszechnie stosuje się inne rozwiązanie (patrz dodatek E) – algorytmy wykorzystywane w aplikacjach działających offline nie sprawdzają się w aplikacjach w czasie rzeczywistego.

Jednak problem interpolacji, a tym samym opisu ruchu w animacji, jest znacznie szerszy. Opis ruchu jest chyba najważniejszym czynnikiem decydującym o odbiorze (akceptacji) animacji przez widza. Norman McLaren, pionier animacji oraz zdobywca Oscara, w autobiografii Chucka Jonesa (wybitnego ilustratora w studio Warner Brothers) powiedział:

„W animacji nie jest sztuką narysować ruszające się obrazki – sztuką jest zobrazić ruch. A ważniejsze od tego, co znajduje się na klatkach jest to, co dzieje się pomiędzy nimi” (ang. Animation is not the art of drawings-that-move, but rather the art of movements-that-are-drawn. What happens between each frame is more important than what happens on each frame). [15]

Wypowiedź ta w bardzo trafny sposób określa problem ruchu i roli jego opisu w animacji. Z jednej strony animacja zbudowana jest z elementów składających się z innych dziedzin sztuki, np. ilustracji czy też malarstwa, ale jest od nich bogatsza w treść. Samo przedstawienie obiektów na statycznym płótnie nie niesie ze sobą pełnej informacji o narysowanych obiektach. W animacji dopiero ruch zdradza ich prawdziwą naturę oraz daje możliwość zrozumienia posiadanych przez nie właściwości fizycznych lub cech emocjonalnych postaci. Z drugiej strony animacja, jak każda sztuka, jest pewnym uproszczeniem i symbolicznym przedstawieniem rzeczywistości. Nawet grafika realistyczna jest tylko środkiem wyrazu stosowanym przez reżysera.

Mimo tego, że praktycznie wszyscy animatorzy podkreślają, że praca nad animacją sprawia im przyjemność i satysfakcję z kreowania animowanego świata, to jednak dopiero odbiorca (widz) decyduje o odbiorze dzieła. To percepcja i odczucia widza decydują o jakości i akceptacji animacji. Kaczor Donald (studio Walta Disneya), Miś Yogi (wytwórnia Hanna-Barbera) czy Koziołek Matołek (Studio Miniatur Filmowych) są postaciami przedstawionymi z różnymi stopniami uproszczenia przez twórców. Każda z tych postaci ma swoich zwolenników, ale też każdy widz zauważy różnice w opisie ruchu tych przykładowych postaci, np. wielbiciele Disneya podkreślają zawsze płynność i „posuwistość” ruchu. Jednak tym, co łączy wszystkie te przykłady, jest pewnego rodzaju naturalność ruchu. Naturalność rozumiana jako zgodność z doświadczeniem widza i jego oczekiwaniem. Widz jest w stanie zaakceptować prawie dowolny poziom uproszczenia (umowności) w rysunku i szczegółach przedstawienia postaci. Trudno mu będzie natomiast zaakceptować nienaturalny ruch postaci. Można zaakceptować prawie dowolne uproszczenia w przedstawieniu postaci animowanego ptaka, ale żeby ten ptak odleciał w odpowiedniej sekwencji filmu, to musi ruszać skrzydłami w sposób odpowiadający ruchom skrzydeł rzeczywistego ptaka.

Problem opisu ruchu był zawsze podstawowym problemem animacji tradycyjnej i staje się jednym z najpoważniejszych problemów współczesnej animacji komputerowej. Motywacją do rozwiązania problemów postawionych w tej pracy jest potrzeba opracowania metod interpolacji ruchu w animacji, metod pozwalających w łatwiejszy sposób kreować trajektorie ruchu przez współczesnych animatorów.

Rozdział 2

Cel, teza i zakres pracy

Celem przeprowadzonych badań było opracowanie nowego sposobu interpolacji parametrów animacji w metodzie klatek kluczowych, przy użyciu krzywej Béziera trzeciego stopnia dla aplikacji czasu rzeczywistego. Kubiczna płaska krzywa Béziera jest wykresem funkcji przestępnej opisującej pewien parametr animacyjny w zależności od czasu.

Analiza problemów wydajnościowo-pamięciowych w aplikacjach czasu rzeczywistego, takich jak gry wideo, upoważnia do postawienia następującej tezy.

„Możliwe jest znalezienie funkcji sklejaney trzeciego stopnia, opisanej przez niewiele kawałków wielomianowych, przybliżającej wspomnianą wyżej funkcję przestępną z dobrą dokładnością w ustalonej metryce i da się to zrobić metodami algebry”.

Określenie „z dobrą dokładnością” rozumiane jest w sposób geometryczny (więcej informacji można znaleźć w rozdziale (4)). Wielomian trzeciego stopnia został wybrany w prezentowanej metodzie ponieważ:

- Wielomian trzeciego stopnia pozwala łączyć krzywe z zachowaniem maksymalnie klasy gładkości C^2 .
- Każdy wielomian co najwyżej trzeciego stopnia można przekształcić do postaci parametrycznej krzywej Béziera trzeciego stopnia. Ta właściwość umożliwi, po aproksymacji zaprezentowaną metodą, podgląd animacji w programach do edycji oraz zapis animacji w dostępnych obecnie formatach plików (takich jak Collada™, FBX® itp), bez dodatkowych koniecznych zmian w tych formatach plików.
- Wyznaczenie wartości wielomianów co najwyżej trzeciego stopnia jest możliwe przy pomocy wzoru $f(x) = ((a*x+b)*x+c)*x+d$. Zastosowanie tego wzoru pozwala sprzętowo wspomagać wyznaczanie wartości wielomianów poprzez trzykrotne użycie specjalnego rozkazu procesora typu FMA (ang. *fused multiply add*)[49]. Rozkaz ten w jednym cyklu wykonuje obliczenia $d = a * b + c$ mnożenia oraz dodawania.
- Minimalizacja liczby N wielomianów użytych do aproksymacji spowoduje, że zapotrze-

bowanie przestrzeni pamięciowej dla przechowywania animacji wyniesie $4 \times N$, ponieważ jednoznaczne przedstawienie wielomianów co najwyżej trzeciego stopnia wymaga przechowania w pamięci komputera czterech współczynników wielomianu.

Do udowodnienia postawionej tezy, przy uwzględnieniu wyżej omówionych założeń, konieczna była realizacja następujących zadań:

- Wyznaczenie krzywej Béziara trzeciego stopnia w parametryzacji, którą można wykorzystać do wyznaczenia aproksymacji w ustalonej metryce funkcji ciągłych.
- Wybór odpowiedniej metryki, w której istnieje ekstremala dla zadanego problemu optymalizacji, natomiast operacje wyznaczania ekstremali byłyby możliwie proste (nie wymagające stosowania zaawansowanych metod numerycznych).
- Znalezienie równania opartego o wyznaczoną ekstremalę, które w sposób jednoznaczny wyznaczy współczynniki wielomianów co najwyżej trzeciego stopnia.
- Przeprowadzenie analizy wartości granicznej błędu aproksymacji, dla których wielomiany aproksymujące będą akceptowalne dla obserwatora.

Praca podzielona została na osiem rozdziałów.

Pierwszy rozdział to wstępny opis pojęcia animacji. Opisane zostały także zasady pozwalające budowanie animacji o pożądanej dynamice. Następnie opisana została metoda klatek kluczowych wykorzystywana w animacji komputerowej oraz właściwości krzywych interpolacyjnych używanych w tej metodzie. Przedstawiona została możliwość konwersji różnego rodzaju funkcji interpolacyjnych do postaci krzywej Béziara trzeciego stopnia. Następnie zarysowany został problem, z jakim programiści aplikacji czasu rzeczywistego (np. gry wideo) zmagają się w interpolowaniu wartości w metodzie klatek kluczowych.

W rozdziale drugim postawiono cel i tezę oraz określono zakres prac do realizacji.

W rozdziale trzecim omówiono szczegółowo krzywą Béziara trzeciego stopnia w najczęściej przedstawianej parametryzacji, która to parametryzacja utrudnia zastosowanie jej bezpośrednio w aplikacjach czasu rzeczywistego. W tym rozdziale zaproponowana została nowa parametryzacja krzywej Béziara trzeciego stopnia jako funkcja zmiany parametru (użytego w metodzie klatek kluczowych) w czasie, oraz przedstawione zostały ograniczenia na punkty kontrolne krzywej Béziara, tak by krzywa ta mogła być wykorzystana jako funkcja interpolacyjna w metodzie klatek kluczowych.

W rozdziale czwartym zaprezentowano metodę jednoznacznego wyznaczenia współczynników wielomianów co najwyżej trzeciego stopnia aproksymujących krzywą Béziara trzeciego stopnia. W tym celu przedstawione zostały warunki zachowania ciągłości w węzłach, jakie prezentowana metoda wymaga oraz dodatkowe warunki, które użytkownik może wybrać do lepszej aproksymacji krzywej Béziara w konkretnym przypadku. Następnie omówiono przestrzenie

unormowane i przedstawiono normy całkowite, wykorzystane do wyznaczenia funkcji minimalizującej. Ekstremum warunkowe tej funkcji minimalizującej pozwala jednoznacznie wyznaczyć współczynniki wielomianów aproksymujących.

Rozdział piąty jest wstępem do analizy zaproponowanej w pracy metody (z rozdziału czwartego). Zawiera opis postaci znormalizowanej krzywej Béziera trzeciego stopnia. W rozdziale tym zaproponowano przykładowe testowe krzywe, które poddane zostaną badaniom i analizie. Opisano także problem wpływu wartości błędu aproksymacji (z rozdziału czwartego) na wizualny odbiór aproksymowanej animacji.

Rozdział szósty zawiera opisy testów, którymi poddana została metoda zaproponowana w pracy. Zaproponowane testy pozwalają zbadać wpływ punktów konfiguracyjnych, użytych w metodzie proponowanej z rozdziału czwartego, na wartość błędu aproksymacji. Dodatkowo przedstawiono testy wydajnościowe i pamięciowe oraz analizę ich wyników na tle wyników innych technik opisanych w dodatku E.

Rozdział siódmy zawiera podsumowanie. Przedstawiono stopień realizacji nakreślonych w pracy zadań oraz uzyskane rezultaty, które potwierdzają słuszność postawionej w pracy tezy.

Rozdział 3

Krzywa Béziara

3.1 Krzywa Béziara jako wykres funkcji

Krzywą Béziara n -tego stopnia [63] nazywamy zbiór punktów płaszczyzny R^2 , wyznaczony przez $n + 1$ jej punktów P_0, P_1, \dots, P_n oraz parametryzację

$$P(t) = \sum_{i=0}^n P_i B_i^n(t) \quad \text{gdzie } t \in [0, 1] \quad (3.1)$$

gdzie $B_i^n(t)$ jest wielomianem stopnia n wyrażonym wzorem

$$B_i^n(t) = \binom{n}{i} t^i (1-t)^{n-i} \quad \text{gdzie } i = 0 \dots n \quad (3.2)$$

Wielomiany (3.2) są blisko związane z wielomianami Bernsteina, i często w publikacjach tak określane w kontekście krzywych Béziara. Pierwotnie wielomiany tej postaci zostały wykorzystane do udowodnienia twierdzenia Weierstrassa dla przypadku jednowymiarowego [2].

Punkty P_0, P_1, \dots, P_n nazywamy punktami kontrolnymi.

W zainteresowaniu niniejszej pracy są tylko krzywe Béziara trzeciego stopnia i wówczas ich parametryzacja ma postać

$$P(t) = P_0(1-t)^3 + 3P_1t(1-t)^2 + 3P_2t^2(1-t) + P_3t^3 \quad \text{gdzie } t \in [0, 1] \quad (3.3)$$

Funkcję wektorową $P(t)$ możemy przedstawić także w postaci

$$P(t) = (x(t), y(t)) \quad (3.4)$$

gdzie $x(t)$ oraz $y(t)$ są wielomianami co najwyżej trzeciego stopnia,

$$\begin{aligned} x(t) &= x_0(1-t)^3 + 3x_1t(1-t)^2 + 3x_2t^2(1-t) + x_3t^3 & t \in [0, 1] \\ y(t) &= y_0(1-t)^3 + 3y_1t(1-t)^2 + 3y_2t^2(1-t) + y_3t^3 & t \in [0, 1] \end{aligned} \quad (3.5)$$

Wzór na parametryzację (w bazie wielomianów Bernsteina) można przedstawić w postaci parametrycznej (w bazie potęgowej)

$$P(t) = \begin{cases} x(t) = g(t) = \alpha_0 t^3 + \beta_0 t^2 + \gamma_0 t + \delta_0 & t \in [0, 1] \\ y(t) = f(t) = \alpha_1 t^3 + \beta_1 t^2 + \gamma_1 t + \delta_1 & t \in [0, 1] \end{cases} \quad (3.6)$$

t należy do odcinka $[0, 1]$.

Przybliżania krzywej w opisie parametrycznym (czyli postać krzywej Béziara ze wzoru (3.6)) prowadzi do konstruowania ciągu par funkcji. Podczas gdy przybliżanie krzywej jako wykresu funkcji sprowadza się do konstruowania ciągu funkcji, co jest najczęściej dużo łatwiejsze od konstruowania par funkcji.

Zatem poszukiwana jest postać krzywej jako wykresu funkcji $y = y(x)$, czyli przejście od opisu krzywej w postaci parametrycznej do opisu krzywej jako wykresu funkcji.

W tym celu należy rozwikłać względem parametru t pierwsze równanie z (3.6), a następnie otrzymaną funkcję $t = t(x)$ podstawić do drugiego równania z (3.6). Złożenie $y(x) = y(t(x))$ zostanie zapisane w prostszej postaci

$$C(x) = f(g^{-1}(x)) \quad (3.7)$$

Funkcje $C(x)$ będziemy nazywać **funkcją wejściową**, funkcję g nazywać będziemy **funkcją wewnętrzną**, funkcję f **funkcją zewnętrzną**.

Ponieważ funkcja g jest wielomianem stopnia trzeciego, zatem jest jednoznacznie wyznaczona przez swoje cztery współczynniki liczbowe a_0, b_0, c_0, d_0 , które mogą być wyliczone przy pomocy wzoru

$$\begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{bmatrix} = \frac{1}{(x_3 - x_0)^3} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3x_3 & -3x_0 - 6x_3 & 6x_0 + 3x_3 & -3x_0 \\ -3x_3^2 & 6x_0x_3 + 3x_3^2 & -3x_0^2 - 6x_0x_3 & 3x_0^2 \\ x_3^3 & -3x_0x_3^2 & 3x_0^2x_3 & -x_0^3 \end{bmatrix} * \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad (3.8)$$

Podobnie funkcja f jest wielomianem stopnia trzeciego, więc wyznaczona przez kolejne cztery współczynniki liczbowe a_1, b_1, c_1, d_1 . Podobnie jak poprzednio współczynniki te można wyliczyć przy pomocy wzoru

$$\begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} = \frac{1}{(x_3 - x_0)^3} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3x_3 & -3x_0 - 6x_3 & 6x_0 + 3x_3 & -3x_0 \\ -3x_3^2 & 6x_0x_3 + 3x_3^2 & -3x_0^2 - 6x_0x_3 & 3x_0^2 \\ x_3^3 & -3x_0x_3^2 & 3x_0^2x_3 & -x_0^3 \end{bmatrix} * \begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad (3.9)$$

Zakresem zmienności x jest odcinek $[x_0, x_3]$ wyznaczony przez pierwsze współrzędne punktów, P_0 i P_3 odpowiednio. Przy wyprowadzaniu powyższych wzorów zostały uwzględnione warunki

$$\begin{cases} C(x_0) = P(0) \\ C(x_3) = P(1) \end{cases} \quad (3.10)$$

Przekształcenia wyznaczone przez macierze znajdujące się w prawych stronach wzorów () są odwracalne, zatem każdą parę wielomianów co najwyżej trzeciego stopnia (o współczynnikach a_0, b_0, c_0, d_0 dla pierwszego wielomianu oraz a_1, b_1, c_1, d_1 dla drugiego) w przedziale $[x_0, x_3]$ można przedstawić następująco.

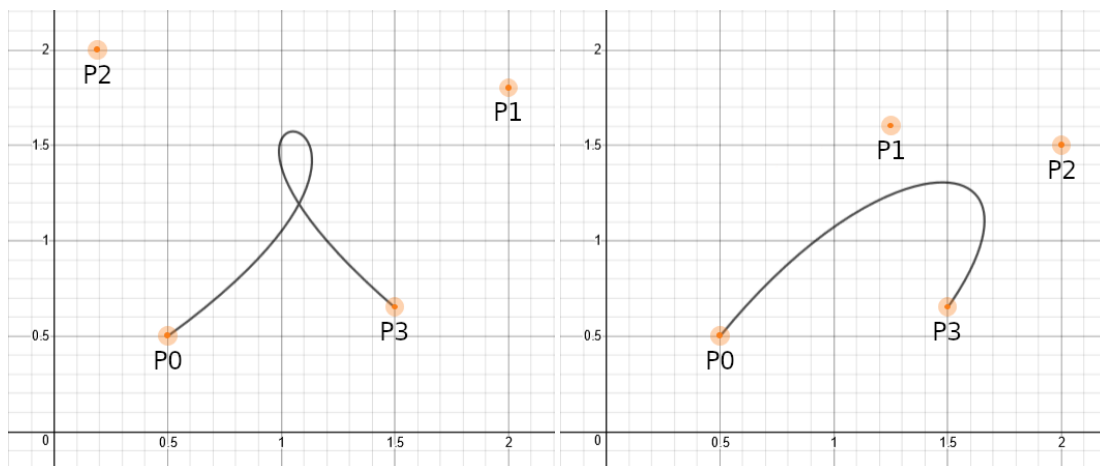
$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} x_0^3 & x_0^2 & x_0 & 1 \\ x_0^2 x_3 & \frac{x_0^2 + 2x_0 x_3}{3} & \frac{2x_0 + x_3}{3} & 1 \\ x_0 x_3^2 & \frac{x_3^2 + 2x_0 x_3}{3} & \frac{x_0 + 2x_3}{3} & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \end{bmatrix} * \begin{bmatrix} a_0 \\ b_0 \\ c_0 \\ d_0 \end{bmatrix} \quad (3.11)$$

oraz

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} x_0^3 & x_0^2 & x_0 & 1 \\ x_0^2 x_3 & \frac{x_0^2 + 2x_0 x_3}{3} & \frac{2x_0 + x_3}{3} & 1 \\ x_0 x_3^2 & \frac{x_3^2 + 2x_0 x_3}{3} & \frac{x_0 + 2x_3}{3} & 1 \\ x_3^3 & x_3^2 & x_3 & 1 \end{bmatrix} * \begin{bmatrix} a_1 \\ b_1 \\ c_1 \\ d_1 \end{bmatrix} \quad (3.12)$$

Z przedstawioną parametryzacją (3.7), wiążą się dwa powiązane ze sobą problemy.

1. Krzywa Béziera w parametryzacji (3.1) (w szczególności krzywa Béziera trzeciego stopnia) może przedstawiać różnego rodzaju trajektorie, nawet takie, które zawierają pętle albo łuki, na przykład takie jak na rysunku 3.1.



Rysunek 3.1: Przykład Krzywych Béziera trzeciego stopnia zawierających pętle oraz łuki.

Takiej postaci trajektorii nie da się zapisać jako wykresu funkcji $y = y(x)$ lub $x = x(y)$, gdyż nie spełniona jest definicja funkcji jako „przyporządkowania każdemu elementowi ze zbioru X dokładnie jednego elementu ze zbioru Y ”. Trajektorie te mogą być opisane przy pomocy funkcji wieloznacznych, w szczególności powierzchni Riemanna.

2. Zagadnienie odwracania funkcji (przekształceń) w problemie opisywanym w pracy zawężone jest do wielomianów i w ogólnym przypadku prowadzi do pojęcia funkcji wieloznacznej [34] (funkcji analitycznej w sensie Weierstrassa [20]). Przechodząc do zmiennej zespolonej i rozpatrując funkcje odwrotne do wielomianów dochodzimy do szczególnej postaci funkcji wieloznacznych: powierzchni Riemanna [34]. Pojęcie to byłoby pożytecznie przy rozpatrywaniu wszystkich postaci krzywych Béziera w parametryzacji (3.1), między innymi takich jak na rysunku 3.1.

Dlatego, aby krzywa Béziera w parametryzacji (3.6) mogła być użyta w poszukiwaniu aproksymacji wielomianami co najwyżej trzeciego stopnia, nie są dopuszczalne wszystkie jej postacie, lecz tylko takie, które spełniają warunek

$$g'(x) > 0, \quad x \in [x_0, x_3] \quad (3.13)$$

Zbiór wszystkich funkcji spełniających ten warunek oznaczone zostanie Ω_+ . Warunek ten (3.13) gwarantuje, że funkcja wewnętrzna będzie funkcją ściśle rosnącą, co skutkuje tym że będzie odwracalna w przedziale $x \in [x_0, x_3]$, oraz funkcja $C(x)$ nie będzie posiadała łuków oraz pętli jak na rysunku 3.1.

Ponadto dopuszczalny jest także słabszy warunek

$$g'(x) \geq 0, \quad x \in [x_0, x_3] \quad (3.14)$$

krzywe $C(x)$, które spełniają ten warunek, dopuszczają istnienie jednego punktu przegięcia w $[x_0, x_3]$ (nie mamy w tym przypadku do czynienia z lokalnym ekstremum, gdyż nie jest spełniony warunek zmiany znaku pierwszej pochodnej funkcji w otoczeniu punktu stacjonarnego, czyli punktu jej zerowania się).

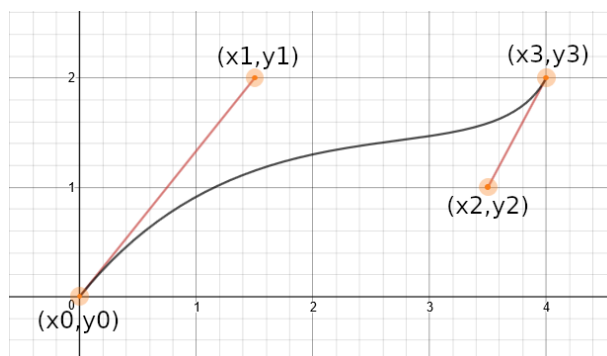
Zbiór wszystkich funkcji spełniających ten warunek (3.14) zostanie oznaczony Ω_0 .

Zbiór $\Omega_1 = \Omega_0 \setminus \Omega_+$, wyznacza rodzinę funkcji których pierwsza pochodna ma jedno miejsce zerowe w przedziale $[x_0, x_3]$.

3.1.1 Interpretacja krzywej Béziera w metodzie klatek kluczowych

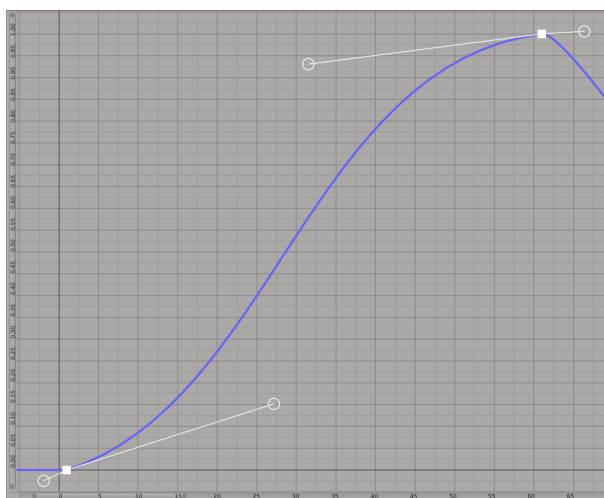
W rozdziale 1.3 (strona 16) zostało przedstawione, że klatka kluczowa to para uporządkowana (x, y) . Punkty kontrolne (P_0, P_1, P_2, P_3) krzywej Béziera, zdefiniowane przez pary współrzędnych $P_n = (x_n, y_n)$ gdzie $n = 0, 1, 2, 3$, w naturalny sposób wkomponowują się w idee metody klatek kluczowych.

Wykorzystanie parametryzacji krzywej w postaci (3.5) w prosty sposób pozwala wykreślić trajektorię na płaszczyźnie R^2 (np. jak na rysunku 3.2).



Rysunek 3.2: Przykładowa krzywa Béziera trzeciego stopnia w R^2 .

Jednak w animacji komputerowej, wykorzystującej metodę klatek kluczowych, parametryzacja taka jest kłopotliwa. Powodem jest sposób, w jaki w animacji komputerowej interpretowana jest jedna z osi.



Rysunek 3.3: Przykładowa krzywa Béziera trzeciego stopnia użyta jako krzywa interpolacyjna w metodzie klatek kluczowych, na podstawie edytora animacji w programie Blender.

Jak pokazano na rysunku 3.3, wartości na osi poziomej reprezentują numery klatek, co w metodzie klatek kluczowych przekłada się na czas w odtwarzanej animacji (odtwarzając klatki animacji po kolei z ustaloną prędkością, np. 24 klatki na sekundę, jak ma to miejsce w kinematografii, numer klatki wyznacza konkretny czas w animacji). Oś pionowa interpretowana jest jako wartość parametru poddawany interpolowaniu.

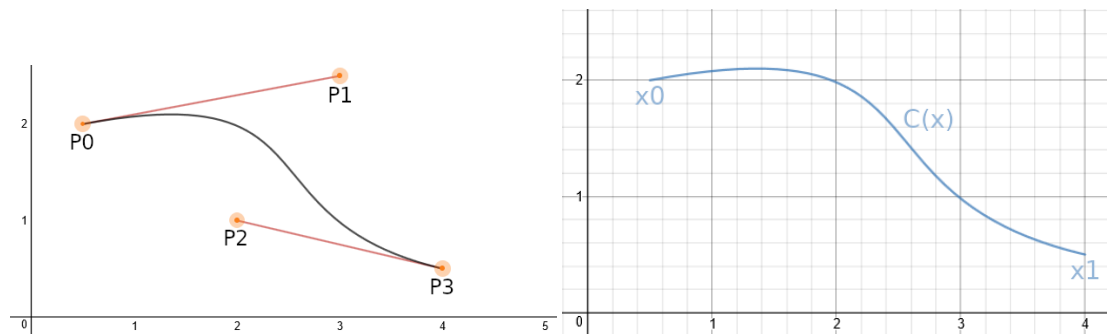
Jak to było przedstawione w listingu Algorytm 2 oraz Algorytm 3, interpolacja w metodzie klatek kluczowych krzywej Béziera w parametryzacji (3.5) polega najpierw na wyznaczeniu parametru t dla zadanej wartości x (czyli rozwiązaniu równania (1.7) metodami algebraicznymi przedstawionymi w listingu Algorytm 3 lub numerycznymi (1.8)), aby następnie tak wyznaczoną wartość wykorzystać jako argument funkcji zewnętrznej.

Takie postępowanie jest konieczne, gdyż krzywa Béziera (w parametryzacji (3.1)) zaprojektowana była jako jedno z narzędzi wykorzystywanych w modelowaniu geometrycznym. Użycie tej samej parametryzacji dla krzywej interpolacyjnej w metodzie klatek kluczowych jest problematyczne, szczególnie dla aplikacji czasu rzeczywistego (np. gier wideo), gdyż parametryzacja (3.5) nie udostępnia wzoru na wartość interpolowanego parametru jako funkcji czasu (gdzie współrzędna x , czyli numer klatki, jest traktowana jako czas animacji, a nie parametr t).

Warto zaznaczyć, że zmienna x mimo, że może wydawać się współrzędną przestrzenną, to w interpretacji animacji metodą klatek kluczowych jest współrzędną czasową.

Wyznaczona postać funkcji $C(x)$ (3.7) pozwala wyznaczyć wartość parametru (animowanego metodą klatek kluczowych) jako funkcji czasu.

Parametryzacje (3.1) oraz (3.7) opisują tę samą krzywą. Różnica jest w postaci użytego wzoru: pierwsza jest w postaci parametrycznej względem zmiennej t , druga jest w postaci jawnej względem zmiennej x . Obie te parametryzacje przedstawia rysunek 3.4.



Rysunek 3.4: Rysunek z lewej strony na którym przedstawiono postać parametryczną krzywej Béziera trzeciego stopnia z widocznymi punktami kontrolnymi. Na rysunku z prawej przedstawiono postać jawną tej samej krzywej jako funkcje czasu.

Parametryzacją krzywej Béziera (3.1) jak pokazano wcześniej umożliwia konstruowanie krzywej takiej jak na rysunku 3.1. W przypadku gdy obie osie w tych przykładach są osiami przestrzennymi, taka postać krzywej Béziera nie powoduje problemów z interpretacją ruchu. Natomiast interpolacja tymi krzywymi nie jest możliwa, gdyż podążająca za nimi interpolacja wymuszałaby, w pewnych przedziałach, poruszanie się „wstecz czasu”. Ten sam problem można opisać jako niedopuszczalną możliwą bilokację (w danej chwili obiekt jest w dwóch różnych miejscach).

Jednak spełnienie warunku (3.13) lub warunku (3.14) pozwala na utożsamienie, w metodzie klatek kluczowych, współrzędnej x z czasem.

3.1.2 Funkcja odwrotna do wielomianu co najwyżej trzeciego stopnia

W celu wyznaczenia wzoru na funkcję $C(x)$, określoną wzorem (3.7), koniecznością staje się znalezienie funkcji odwrotnej do wielomianu co najwyżej trzeciego stopnia. Rozpatrzmy zatem dowolny wielomian co najwyżej trzeciego stopnia,

$$y = W(x) = ax^3 + bx^2 + cx + d \quad (3.15)$$

Równość (3.15) trzeba rozwikłać względem zmiennej x . Sprowadza się to do rozwiązania problemu znajdowania pierwiastków wielomianów co najwyżej trzeciego stopnia.

Zaznaczyć trzeba także iż problem znajdowania pierwiastków ograniczony jest tylko do rzeczywistych parametrów wielomianu oraz dziedziny funkcji w zbiorze liczb rzeczywistych (gdyż w tej dziedzinie wykorzystywany jest w animacji komputerowej).

By rozpatrywany wielomian (3.15), mógł być wykorzystany w problemach opisywanych w pracy, należy rozpatrywać go wraz z warunkiem (3.13). Możliwych jest kilka przypadków w zależności od stopnia wielomianu. Ponieważ wielomiany stopnia 0 są funkcjami stałymi, czyli nie spełniają warunku (3.13), nie będą tu rozpatrywane.

- $W(x)$ jest pierwszego stopnia $W(x) = cx + d$.

Funkcja tej postaci spełnia warunek (3.13) wtedy i tylko wtedy gdy

$$c > 0. \quad (3.16)$$

- $W(x)$ jest drugiego stopnia: $W(x) = bx^2 + cx + d$.

Warunek (3.13) oznacza że

$$2bx + c > 0 \iff (b > 0 \wedge x > -\frac{c}{2b}) \vee (b < 0 \wedge x < -\frac{c}{2b}). \quad (3.17)$$

- $W(x)$ trzeciego stopnia: $W(x) = ax^3 + bx^2 + cx + d$.

Funkcja tej postaci spełnia warunek (3.13), $3ax^2 + 2bx + c > 0$ wtedy i tylko wtedy gdy

$$(a > 0 \wedge c > \frac{b^2}{3a}) \vee (a > 0 \wedge c < \frac{b^2}{3a} \wedge x > \frac{\sqrt{b^2 - 3ac} - b}{3a}). \quad (3.18)$$

A zatem wzory na pierwiastki wielomianu (3.15) są następujące:

- Dla wielomianu 1-go stopnia

$$W(x) = cx + d = 0 \Leftrightarrow$$

$$x = \frac{-d}{c}$$

- Dla wielomianu 2-go stopnia

$$W(x) = bx^2 + cx + d = 0 \Leftrightarrow$$

$$x_0 = \frac{-c - \sqrt{c^2 - 4bd}}{2b}$$

$$x_1 = \frac{-c + \sqrt{c^2 - 4bd}}{2b}$$

- Dla wielomianu trzeciego stopnia

W tym przypadku konieczne jest przekształcenie wielomianu trzeciego stopnia do postaci kanonicznej

$$W(x) = ax^3 + bx^2 + cx + d = 0$$

$$x = s - \frac{b}{3a}$$

$$p = \frac{3ac - b^2}{3a^2}$$

$$q = \frac{2b^3 - 9abc + 27a^2d}{27a^3}$$

$$W(t) = t^3 + pt + q$$

po czym w zależności od wartości p oraz q wzory na pierwiastki wielomianu trzeciego stopnia przedstawiają się następująco:

– $p = 0$

$$s_0 = \sqrt[3]{-q}$$

$$x_0 = \sqrt[3]{-q} - \frac{b}{3a}$$

– $p > 0$

$$s_0 = -2\sqrt{\frac{p}{3}} \sinh\left(\frac{1}{3} \operatorname{arcsinh}\left(\frac{3q}{2p}\sqrt{\frac{3}{p}}\right)\right)$$

$$x_0 = -2\sqrt{\frac{p}{3}} \sinh\left(\frac{1}{3} \operatorname{arcsinh}\left(\frac{3q}{2p}\sqrt{\frac{3}{p}}\right)\right) - \frac{b}{3a}$$

– $p < 0 \wedge 4p^3 + 27q^2 > 0$

$$s_0 = -2 \operatorname{sgn}(q) \sqrt{-\frac{p}{3}} \cosh\left(\frac{1}{3} \operatorname{arccosh}\left(-\frac{3|q|}{2p}\sqrt{-\frac{3}{p}}\right)\right)$$

$$x_0 = -2 \operatorname{sgn}(q) \sqrt{-\frac{p}{3}} \cosh\left(\frac{1}{3} \operatorname{arccosh}\left(-\frac{3|q|}{2p}\sqrt{-\frac{3}{p}}\right)\right) - \frac{b}{3a}$$

$$-p < 0 \wedge 4p^3 + 27q^2 < 0$$

$$s_0 = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right)\right)$$

$$s_1 = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right) - \frac{2\pi}{3}\right)$$

$$s_2 = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right) - \frac{4\pi}{3}\right)$$

$$x_0 = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right)\right) - \frac{b}{3a}$$

$$x_1 = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right) - \frac{2\pi}{3}\right) - \frac{b}{3a}$$

$$x_2 = 2\sqrt{-\frac{p}{3}} \cos\left(\frac{1}{3} \arccos\left(\frac{3q}{2p}\sqrt{-\frac{3}{p}}\right) - \frac{4\pi}{3}\right) - \frac{b}{3a}$$

Wykorzystując powyższe wzory na pierwiastki wielomianów co najwyżej trzeciego stopnia, otrzymujemy wzory na funkcje do nich odwrotne $W^{-1}(x)$.

- Dla wielomianu 1-go stopnia

$$W(x) = cx + d$$

$$l = \frac{1}{c}$$

$$k = -\frac{d}{c}$$

$$W^{-1}(x) = lx + k$$

- Dla wielomianu 2-go stopnia

$$W(x) = bx^2 + cx + d$$

$$k = \frac{c^2}{4b^2} - \frac{d}{b}$$

$$l = \frac{1}{b}$$

$$B = -\frac{d}{2b}$$

$$W_0^{-1}(x) = B - \sqrt{lx + k}$$

$$W_1^{-1}(x) = B + \sqrt{lx + k}$$

- Dla wielomianu trzeciego stopnia

$$W(x) = ax^3 + bx^2 + cx + d$$

$$p = \frac{3ac - b^2}{3a^2}$$

$$q = \frac{2b^3 - 9abc + 27a^2d}{27a^3}$$

– $p = 0$

$$k = -\frac{2b^3 - 9abc + 27a^2d}{27a^3}$$

$$l = \frac{1}{a}$$

$$B = -\frac{b}{3a}$$

$$W^{-1}(x) = \sqrt[3]{lx + k} + B$$

– $p > 0$

$$A = -2\sqrt{\frac{p}{3}}$$

$$B = -\frac{b}{3a}$$

$$k = \frac{3}{-Ap} \frac{2b^3 - 9abc + 27a^2d}{27a^3}$$

$$l = -\frac{3}{-Ap} \frac{1}{a}$$

$$W^{-1}(x) = A \sinh\left(\frac{1}{3} \operatorname{arcsinh}(lx + k)\right) + B$$

– $p < 0 \wedge 4p^3 + 27q^2 > 0$

$$A = -2\sqrt{-\frac{p}{3}}$$

$$B = -\frac{b}{3a}$$

$$k = \frac{3}{Ap} \frac{(2b^3 - 9abc + 27a^2d)}{27a^3}$$

$$l = -\frac{3}{Ap} \frac{1}{a}$$

$$W^{-1}(x) = \operatorname{sgn}(lx + k)A \cosh\left(\frac{1}{3} \operatorname{arccosh}(|lx + k|)\right) + B$$

$$- p < 0 \wedge 4p^3 + 27q^2 < 0$$

$$A = 2\sqrt{-\frac{p}{3}}$$

$$B = -\frac{b}{3e}$$

$$k = \frac{3}{Ap} \frac{(2b^3 - 9abc + 27a^2d)}{27a^3}$$

$$l = \frac{3}{Ap} - \frac{1}{a}$$

$$W_0^{-1}(x) = A \cos\left(\frac{1}{3} \arccos(lx + k)\right) + B$$

$$W_1^{-1}(x) = A \cos\left(\frac{1}{3} \arccos(lx + k) - \frac{2\pi}{3}\right) + B$$

$$W_2^{-1}(x) = A \cos\left(\frac{1}{3} \arccos(lx + k) - \frac{4\pi}{3}\right) + B$$

Otrzymane wyżej wzory pozwalają na skonstruowanie funkcji $g^{-1}(x)$ odwrotnej do funkcji g potrzebnej we wzorze (3.7).

3.1.3 Funkcja $C(x)$

Mając wyznaczone wzory na funkcje $g^{-1}(x)$, możliwe jest przedstawienie poszukiwanej parametryzacji (3.7) jako wykresu funkcji.

- $a_1 = 0 \wedge b_1 = 0$

$$\begin{aligned}
C(x) &= \alpha x^3 + \beta x^2 + \gamma x + \delta \\
l &= \frac{1}{c_1} \\
k &= -\frac{d_1}{c_1} \\
\alpha &= a_0 l^3 \\
\beta &= 3a_0 l^2 k + b_0 l^2 \\
\gamma &= 3a_0 l k^2 + 2b_0 l k + c_0 l \\
\delta &= a_0 k^3 + b_0 k^2 + c_0 k + d_0
\end{aligned} \tag{3.19}$$

• $a_1 = 0$

$$\begin{aligned}
C(x) &= \alpha \Phi(x)^3 + \beta \Phi(x)^2 + \gamma \Phi(x) + \delta \\
k &= \frac{c_1^2}{4b_1^2} - \frac{d_1}{b_1} \\
l &= \frac{1}{b_1} \\
B &= -\frac{d_1}{2b_1} \\
\Phi(x) &= \sqrt{lx + k} \\
\alpha &= \text{sgn}(b_0) * a_0 \\
\beta &= 3a_0 B + b_0 \\
\gamma &= \text{sgn}(b_0) * (3a_0 B^2 + 2b_0 B + c_0) \\
\delta &= a_0 B^3 + b_0 B^2 + c_0 B + d_0
\end{aligned} \tag{3.20}$$

• $a_1 \neq 0 \wedge p = 0$

$$\begin{aligned}
C(x) &= \alpha \Phi^3(x) + \beta \Phi^2(x) + \gamma \Phi(x) + \delta \\
B &= -\frac{b}{3a_1} \\
k &= \frac{2b_1^3 - 9a_1 b_1 c_1 + 27a_1^2 d_1}{27a_1^3} \\
l &= \frac{1}{a_1} \\
\Phi(x) &= \sqrt[3]{lx + k} \\
\alpha &= a_0 \\
\beta &= 3a_0 B + b_0 \\
\gamma &= 3a_0 B^2 + 2b_0 B + c_0 \\
\delta &= a_0 B^3 + b_0 B^2 + c_0 B + d_0
\end{aligned} \tag{3.21}$$

- $a_1 \neq 0 \wedge p > 0$

$$\begin{aligned}
C(x) &= \alpha\Phi^3(x) + \beta\Phi^2(x) + \gamma\Phi(x) + \delta \\
A &= -2\sqrt{\frac{p}{3}} \\
B &= -\frac{b}{3a_1} \\
k &= \frac{3}{-Ap} \frac{2b_1^3 - 9a_1b_1c_1 + 27a_1^2d_1}{27a_1^3} \\
l &= -\frac{3}{-Ap} \frac{1}{a_1} \\
\Phi(x) &= \sinh\left(\frac{1}{3} \operatorname{arcsinh}(lx + k)\right) \\
\alpha &= a_0A^3 \\
\beta &= 3a_0A^2B + b_0A^2 \\
\gamma &= 3a_0AB^2 + 2b_0AB + c_0A \\
\delta &= a_0B^3 + b_0B^2 + c_0B + d_0
\end{aligned} \tag{3.22}$$

- $a_1 \neq 0 \wedge p < 0 \wedge 4p^3 + 27q^2 > 0$

$$\begin{aligned}
C(x) &= \alpha\Phi^3(x) + \beta\Phi^2(x) + \gamma\Phi(x) + \delta \\
A &= -2\sqrt{-\frac{p}{3}} \\
B &= -\frac{b}{3a_1} \\
k &= \frac{3}{Ap} \frac{(2b_1^3 - 9a_1b_1c_1 + 27a_1^2d_1)}{27a_1^3} \\
l &= -\frac{1}{a_1} \frac{3}{Ap} \\
\Phi(x) &= \cosh\left(\frac{1}{3} \operatorname{arccosh}(|lx + k|)\right) \\
\alpha &= \operatorname{sgn}(q) * a_0A^3 \\
\beta &= 3a_0A^2B + b_0A^2 \\
\gamma &= \operatorname{sgn}(q) * (3a_0AB^2 + 2b_0AB + c_0A) \\
\delta &= a_0B^3 + b_0B^2 + c_0B + d_0
\end{aligned} \tag{3.23}$$

- $a_1 \neq 0 \wedge p < 0 \wedge 4p^3 + 27q^2 < 0$

$$\begin{aligned}
C_u(x) &= \alpha\Phi_u^3(x) + \beta\Phi_u^2(x) + \gamma\Phi_u(x) + \delta \\
u &\in \{0, 1, 2\} \\
A &= 2\sqrt{-\frac{p}{3}} \\
B &= -\frac{b}{3a_1} \\
k &= \frac{3}{Ap} \frac{(2b_1^3 - 9a_1b_1c_1 + 27a_1^2d_1)}{27a_1^3} \\
l &= -\frac{1}{a_1} \frac{3}{Ap} \\
\Phi_u(x) &= \cos\left(\frac{1}{3} \arccos(lx + k) - u\frac{2\pi}{3}\right) \\
\alpha &= a_0A^3 \\
\beta &= 3a_0A^2B + b_0A^2 \\
\gamma &= 3a_0AB^2 + 2b_0AB + c_0A \\
\delta &= a_0B^3 + b_0B^2 + c_0B + d_0
\end{aligned} \tag{3.24}$$

Poszukiwana parametryzacja okazała się składać z sześciu rodzin funkcji. O ich postaci decydują wartości współczynników funkcji wewnętrznej g . W ogólnym przypadku wielomianu trzeciego stopnia mamy do czynienia z czterema współczynnikami, przybierającymi wartości w zbiorze liczb rzeczywistych. Oznacza to, że wielomiany te tworzą czterowymiarową przestrzeń wektorową. Już to częściowo wyjaśnia zróżnicowanie postaci funkcji do nich odwrotnych, a co za tym idzie postaci poszukiwanej funkcji $C(x)$.

W przypadku gdy $a_1 = 0 \wedge b_1 = 0$ (czyli g jest funkcją liniową) funkcja $C(x)$ jest wielomianem co najwyżej trzeciego stopnia.

W pozostałych przypadkach otrzymujemy złożenia wielomianu z funkcjami przestępnymi.

Rozdział 4

Nowy sposób aproksymacji krzywych Béziara trzeciego stopnia

Pojęcie aproksymacji czyli inaczej „przybliżanie” wymaga wyjaśnienia, w jaki sposób rozumiemy występującą tu „bliskość”. Najczęściej „bliskość” rozumiana jest w sposób geometryczny. Zatem wymagane jest wprowadzenie pojęcia odległości. W szeroko rozumianej geometrii odległość mierzona jest przy pomocy metryki. Dlatego w tym punkcie konieczne jest wyjaśnienie, w jaki sposób mierzony będzie błąd aproksymacji krzywej Béziara. Warto także wyjaśnić, że pojęcie błędu aproksymacji (lub dokładności aproksymacji) funkcji rozumie się jako odległość funkcji aproksymowanej do funkcji aproksymującej w ustalonej metryce przestrzeni funkcji ciągłych.

4.1 Opis metody

Metoda prezentowana w pracy dla danych $N + 1$ punktów $s_0, s_1, \dots, s_{N-1}, s_N$, które nazwano węzłami, a dodatkowo s_0, s_N nazywano **węzłami zewnętrznymi**, podczas gdy węzły s_1, \dots, s_{N-1} - **węzłami wewnętrznymi**, definiuje N wielomianów co najwyżej trzeciego stopnia

$f_0(x), f_1(x), \dots, f_{N-1}(x)$, takich że

$$C(x) \approx \sum_{i=0}^{N-1} f_i(x) \quad x \in [x_0, x_3] \quad (4.1)$$

gdzie

$$f_i(x) = \begin{cases} a_i x^3 + b_i x^2 + c_i x + d_i & \text{dla } x \in [s_i, s_{i+1}] \\ 0 & \text{dla } x \notin [s_i, s_{i+1}] \end{cases} \quad (4.2)$$

a węzły s_0, s_N są końcami przedziału określoności funkcji wejściowej $C(x)$

$$\begin{cases} s_0 = x_0 \\ s_N = x_3 \end{cases} \quad (4.3)$$

Metoda prezentowana w pracy polega na wyznaczeniu w węzłach, zewnętrznych oraz wewnętrznych, warunków ciągłości wielomianów (opisane w rozdziale 4.2), oraz wyznaczeniu funkcji minimalizującej z wykorzystaniem normy przestrzeni unormowanej (opisanych w rozdziale 4.3.4), do jednoznacznego wyznaczenia wszystkich współczynników wielomianów (4.1).

4.2 Warunki ciągłości w węzłach

Prezentowana metoda dzieli warunki ciągłości w węzłach na dwie kategorie: warunki ciągłości konieczne oraz opcjonalne.

4.2.1 Warunki konieczne

Warunki konieczne są niezbędne w prezentowanej metodzie i nie można ich pominąć, gdyż funkcje muszą zachować ciągłość C^0 , aby mogły być używane jako funkcje interpolacyjne w programach graficznych. Określono je w następujący sposób:

- Wartość wielomianów aproksymujących w węzłach zewnętrznych powinna być taka sama, jak wartość funkcji wejściowej $C(x)$ w tych węzłach. Jest to wyrażone wzorem:

$$\begin{cases} f_0(s_0) = C(s_0) \\ f_{N-1}(s_N) = C(s_N) \end{cases} \quad (4.4)$$

- Wielomiany (4.1) muszą zachowywać ciągłość C^0 w węzłach wewnętrznych. Ten warunek może zostać spełniony na dwa sposoby:

– „Na sposób aproksymujący”, gdy sąsiadujące funkcje spełniają równanie

$$f_i(s_{i+1}) = f_{i+1}(s_{i+1}), \quad i = 0, 1, \dots, N-3, N-2 \quad (4.5)$$

– „Na sposób interpolujący”, gdy sąsiadujące funkcje spełniają równanie

$$f_i(s_{i+1}) = f_{i+1}(s_{i+1}) = C(s_{i+1}), \quad i = 0, 1, \dots, N-3, N-2 \quad (4.6)$$

W tym przypadku wielomiany $f_i(t)$ można nazwać wielomianami interpolacyjnymi funkcji wejściowej $C(x)$, gdyż ich wartości w węzłach są zgodne i równe odpowiednim wartościom funkcji wejściowej $C(x)$.

W dalszej części pracy dla uproszczenia terminologii wielomiany (4.1) zawsze będą nazywane **wielomianami aproksymującymi** (bez względu na wybrany warunek (4.5) lub (4.6)).

Narzucenie warunków koniecznych implikuje zachowanie ciągłości C^0 w węzłach wewnętrznych (co zostało pokazane jawnie w warunków (4.5) lub (4.6)) oraz zewnętrznych (sąsiadujące

krzywe Béziera trzeciego stopnia spełniające warunek (4.4) zachowują ciągłość C^0 z sąsiadami), co jest koniecznym minimum by:

- Wynik prezentowanej metody, czyli zbiór wielomianów aproksymujących funkcje interpolacyjne krzywej Béziera trzeciego stopnia, możliwy był do zapisu w obecnych plikach wymiany danych (takich jak FBX[®], Collada[™], itp) między aplikacjami graficznymi (takimi jak Blender, 3ds Max oraz Maya).
- Zachowanie ciągłości C^0 pozwoli na odtwarzanie animacji, po zastosowaniu metody opisanej w pracy, w programach graficznych bez konieczności dokonywania ich modyfikacji, gdyż wszystkie wspomniane programy graficzne obsługują tylko takie funkcje.

Narzucenie na funkcje aproksymujące warunków koniecznych powoduje, że są to funkcje sklepane (tzn. kawałkami wielomianowe) klasy C^0 . W dalszej części pracy będzie rozważana aproksymacja wielomianami co najwyżej trzeciego stopnia.

4.2.2 Warunki opcjonalne

Metoda zaproponowana w niniejszej pracy jest elastyczna i pozwala użytkownikom wybrać dodatkowe opcjonalne warunki ciągłości wielomianów w węzłach wewnętrznych oraz zewnętrznych. W ten sposób umożliwi zastosowanie prezentowanej metody w szerszej liczbie możliwych zastosowań.

Warunki opcjonalne można podzielić na następujące przypadki:

- Zachowanie gładkości klasy C^1 połączeń wielomianów w węzłach wewnętrznych.

$$f'_i(s_i) = f'_{i+1}(s_i), \quad i = 1, 2, \dots, N-3, N-2 \quad (4.7)$$

- Zachowanie gładkości klasy C^2 połączeń wielomianów w węzłach wewnętrznych.

$$f''_i(s_i) = f''_{i+1}(s_i), \quad i = 1, 2, \dots, N-3, N-2 \quad (4.8)$$

- Przypisane wartości pierwszej pochodnej wielomianów aproksymujących w węzłach zewnętrznych jako wartości pierwszej pochodnej funkcji wejściowej $C(x)$ w tych węzłach.

$$\begin{cases} f'_0(s_0) = C'(s_0) \\ f'_{N-1}(s_N) = C'(s_N) \end{cases} \quad (4.9)$$

- Przypisane wartości drugiej pochodnej wielomianów aproksymujących w węzłach zewnętrznych jako wartości drugiej pochodnej funkcji $C(x)$ w tych węzłach.

$$\begin{cases} f''_0(s_0) = C''(s_0) \\ f''_{N-1}(s_N) = C''(s_N) \end{cases} \quad (4.10)$$

- Przypisane wartości pierwszej pochodnej wielomianów aproksymujących $f_0(x), \dots, f_{N-1}(x)$ takiej samej jak wartość pierwszej pochodnej funkcji wejściowej $C(x)$ w węzłach wewnętrznych.

$$\begin{cases} f'_i(s_{i+1}) = C'(s_{i+1}), & i = 0, 1, \dots, N-3, N-2 \\ f'_{i+1}(s_{i+1}) = C'(s_{i+1}), & i = 0, 1, \dots, N-3, N-2 \end{cases} \quad (4.11)$$

- Przypisane wartości drugiej pochodnej wielomianów aproksymujących $f_0(x), \dots, f_{N-1}(x)$ takiej samej jak wartość drugiej pochodnej funkcji wejściowej $C(x)$ w węzłach wewnętrznych.

$$\begin{cases} f''_i(s_{i+1}) = C''(s_{i+1}), & i = 0, 1, \dots, N-3, N-2 \\ f''_{i+1}(s_{i+1}) = C''(s_{i+1}), & i = 0, 1, \dots, N-3, N-2 \end{cases} \quad (4.12)$$

Oczywiście prezentowana metoda nie narzuca wyboru kombinacji w wykorzystaniu opcjonalnych warunków dodatkowych. Faktem jest, że niektóre z kombinacji mogą powodować, iż liczba równań ograniczających będzie większa niż liczba poszukiwanych parametrów. Na ogół prowadzi to do układów nadokreślonych, które często nie mają rozwiązania.

Autor pracy zdaje sobie sprawę, że wykorzystanie odpowiedniej kombinacji warunków opcjonalnych wraz z warunkiem (4.3) powoduje, że liczba warunków równa jest liczbie poszukiwanych niewiadomych (współczynników wielomianów co najwyżej trzeciego stopnia) i prezentowana metoda wyznacza te same warunki jak metoda interpolacji funkcjami sklejanymi trzeciego stopnia [23].

Użytkownik ma pełną dowolność wyboru liczby oraz rodzaju użytych warunków opcjonalnych. Trzeba jednak zaznaczyć, że wraz ze wzrostem liczby wszystkich warunków (koniecznych oraz opcjonalnych) maleje liczba stopni swobody w dobieraniu wartości współczynników wielomianów aproksymujących. Interpolacja funkcjami sklejanymi powoduje, że istnieje tylko jedno rozwiązanie, które spełnia wszystkie warunki narzucone przez tę metodę.

4.3 Przestrzenie unormowane. Normy całkowite

W poszukiwaniu odpowiedniej przestrzeni unormowanej, której norma będzie przydatna do zdefiniowania poszukiwanej funkcji minimalizującej, należy najpierw odpowiedzieć na pytanie, „jaki rodzaj błędu aproksymacji będzie najbardziej odpowiedni w problemie aproksymacji funkcji interpolacyjnej w metodzie klatek kluczowych”?

4.3.1 Norma supremum

Dla funkcji f określonej na zbiorze Ω i przybierającej wartości rzeczywiste, norma supremum określona jest wzorem

$$\|f\|_\infty = \sup_{x \in \Omega} |f(x)| \quad (4.13)$$

gdzie $|f(x)|$ oznacza wartość bezwzględną (moduł) liczby $f(x)$. Nietrudno zauważyć że jeśli ciąg funkcyjny f_n zbiega w normie supremum do pewnej funkcji f , to znaczy

$$\lim_{n \rightarrow \infty} \|f_n - f\|_\infty = 0 \quad (4.14)$$

to wówczas dla każdego $x \in \Omega$

$$\lim_{n \rightarrow \infty} f_n(x) = f(x) \quad (4.15)$$

Inaczej mówiąc, zbieżność w normie supremum, zwana zbieżnością jednostajną, implikuje zbieżność punktową. Implikacja odwrotna nie zachodzi, zbieżności te nie są równoważne. Zatem można się pokusić o stwierdzenie, że minimalizacja normy supremum dałaby najlepszą miarę poprawności aproksymacji w problemach aproksymacji animacji. Przez najlepszą miarę rozumie się najłatwiejszy sposób w interpretowania obliczonej wielkości dokładności aproksymowania. Wartość ta, w tym przypadku normy supremum, wyraża największą bezwzględną różnicę między trajektoriami. W przeciwieństwie do norm całkowych, dokładność w tej normie związania jest z percepcją ruchu (więcej informacji na ten temat zawartych jest w rozdziale 5.2).

Wyznaczenie najlepszej, w sensie tej normy, aproksymacji jest jednak problemem nietrywialnym. Wymaga bowiem obliczenia bezwzględnej różnicy pomiędzy wartościami funkcji w każdym punkcie przedziału ich określoności. W rozpatrywanych w pracy przypadkach, poszukiwana bezwzględna różnica może być różnicą wielomianu co najwyżej trzeciego stopnia a funkcją przestępną. Wówczas szukanie supremum różnicy może prowadzić do równań nierozwiązywalnych metodami algebraicznymi. Z kolei zastosowanie metod przybliżonych może prowadzić do utraty kontroli nad rzeczywistym błędem aproksymacji.

4.3.2 Przestrzeń L^p

Przestrzeń $L^p(a, b)$, dla $p > 0$, $a < b$ jest przestrzenią wektorową funkcji określonych na odcinku a, b dla których całka (w sensie Lebesque'a)

$$\int_a^b |f(x)|^p dx \quad (4.16)$$

istnieje i jest skończona. Odległość w $L^p(a, b)$ funkcji f do g jest określona wzorem

$$\rho(f, g) = \left(\int_a^b |f(x) - g(x)|^p dx \right)^{\frac{1}{p}} \quad (4.17)$$

Przestrzeń $L^p(a, b)$ jest przestrzenią unormowaną zupełną (przestrzenią Banacha). Norma funkcji f jest zadana wzorem

$$\|f\|_p = \left(\int_a^b |f(x)|^p dx \right)^{\frac{1}{p}} \quad (4.18)$$

Co więcej, gdy $p = 2$, wówczas przestrzeń $L^2(a, b)$ z tak określoną normą jest równocześnie przestrzenią Hilberta. Iloczyn skalarny przestrzeni $L^2(a, b)$ jest określany wzorem

$$\langle f, g \rangle = \int_a^b f(x)g(x)dx \quad (4.19)$$

Naturalnym środowiskiem dla wielu zagadnień optymalizacji (zagadnień rachunku wariacyjnego) są właśnie przestrzenie Hilberta [42]. Norma L^2 jest jednostajnie wypukła [30]. Implikuje to jednoznaczność rozwiązania dla wielu zagadnień optymalizacji.

Norma (4.18) nie daje możliwości oszacowania normy supremum danej funkcji. Co więcej, zbieżność ciągu funkcyjnego $\{f_n\}$ w L^p i $p < \infty$, w ogólności nie implikuje zbieżności ciągu w normie supremum. Oznacza to, że w sensie normy L^p funkcje f i g mogą być blisko $\|f - g\|_p < \varepsilon$ podczas gdy norma supremum ich różnicy może być dowolnie duża.

Postać definicji normy w przestrzeni $L^1(a, b)$ powoduje, że problem minimalizacyjny prowadzi do problemu znalezienia miejsc zerowych funkcji $C(x) - (ax^3 + bx^2 + cx + d)$, co nie jest możliwe do przeprowadzenia prostymi metodami algebraicznymi.

Natomiast, gdy $p \neq 2$, problem minimalizacji normy $L^p(a, b)$ wiąże się z rozwiązywaniem układu równań nieliniowych, co wymusza stosowanie metod numerycznych i może powodować utratę pełnej kontroli nad błędem aproksymacji. Z tego powodu dla $p \neq 2$ przestrzenie te nie są rozpatrywane w niniejszej pracy.

4.3.3 Przestrzenie Sobolewa

Aproksymacja w normie L^2 nie jest w wielu przypadkach najlepsza. Należy zatem rozważyć silniejszą normę, zachowując przy tym warunek prostoty wyznaczania wielomianów aproksymujących, a przy tym dającym możliwość oszacowania normy supremum. Warunek taki spełnia norma Sobolewa [1]:

$$\|f\|_{1,2} = \|f\|_2 + \|f'\|_2, \quad (4.20)$$

gdzie f' oznacza pochodną funkcji f .

Wprowadzając tę normę Sobolew rozpatrywał funkcje słabo różniczkowalne [39]. Funkcje występujące w niniejszej pracy (z rozdziału 3.1.3) mają zwykłe pochodne, poza być może skończoną liczbą punktów. Nie ma więc potrzeby korzystania z tak ogólnego pojęcia jakim jest słaba pochodna (inaczej pochodna w sensie Sobolewa), ani też pełnego definiowania przestrzeni Sobolewa typu $W^{k,p}$ (przestrzeni funkcji mających słabe pochodne rzędu k całkowalne w p -tej potęgze).

Klasyczna nierówność Sobolewa [1] dla funkcji f jednej zmiennej, określonej na odcinku (a, b) , należącej do $L^2(a, b)$ i której pierwsza pochodna f' jest również funkcją całkowalną z kwadratem, implikuje ciągłość funkcji f w sensie Höldera [32] z wykładnikiem $\gamma = \frac{1}{2}$. Ponadto

$$\|f\|_{C^{0, \frac{1}{2}}(a, b)} \leq K \|f\|_{1, 2}, \quad (4.21)$$

gdzie stała K zależy tylko od długości przedziału $|a - b|$ natomiast

$$\|f\|_{C^{0, \frac{1}{2}}(a, b)} = \|f\|_{\infty} + \sup_{x, y \in (a, b)} \frac{|f(x) - f(y)|}{|x - y|^{\frac{1}{2}}} \quad (4.22)$$

W przypadku odcinka (a, b) stała K wynosi

$$K = ((\sqrt{b - a} + 1)^2 + (b - a)^{-1})^{\frac{1}{2}} \quad (4.23)$$

Więcej informacji o wyprowadzaniu wzoru opisującego stałą K znajduje się w dodatku F.

W przypadku $\gamma = \frac{1}{2}$ nierówność (4.21) implikuje nierówność (4.24)

$$\forall_{x, y \in (a, b)} |f(x) - f(y)| \leq |b - a|^{\frac{1}{2}} K \|f\|_{1, 2}, \quad (4.24)$$

która jest oszacowaniem wzrostu funkcji f na (a, b) w terminach normy Sobolewa. W przypadku samej normy L^2 nie ma takiego oszacowania.

W prezentowanej metodzie zaproponowano normę Sobolewa w przestrzeni $W^{1, 2}(a, b)$ do wyznaczenia minimalizatora aproksymacji funkcji wejściowej $C(x)$, ponieważ daje ona możliwość oszacowania błędu aproksymacji mierzonego w normie supremum. W tym miejscu trzeba zaznaczyć, że norma $W^{1, 2}$ stosowana jest tylko i wyłącznie dla funkcji rodziny Ω_+ , gdyż tylko dla tych funkcji możliwe jest wyznaczenie całek w całym przedziale wejściowym (3.7). Dla funkcji rodziny Ω_1 będzie stosowana norma L^2 .

4.3.4 Funkcja celu

Proponowana metoda pozwala dodatkowo wyznaczyć funkcję celu, której minimalizacja dostarczy poszukiwane rozwiązanie, tj optymalne wielomiany aproksymujące.

$$Z(\mathbf{v}) : \mathbb{R}^{4N} \Rightarrow \mathbb{R} \quad (4.25)$$

gdzie N to liczba wielomianów co najwyżej trzeciego stopnia, a wektor \mathbf{v} zbudowany jest ze współczynników wielomianów aproksymujących.

$$\mathbf{v} = [a_0, b_0, c_0, d_0, \dots, a_{N-1}, b_{N-1}, c_{N-1}, d_{N-1}] \quad (4.26)$$

Funkcja minimalizująca $Z(\mathbf{v})$ wyznaczana jest z wykorzystaniem kwadratu normy $W^{1, 2}$

$$Z(\mathbf{v}) = \sum_{i=0}^{N-1} \left(\int_{s_i}^{s_{i+1}} (C(x) - f_i(x))^2 dx + \int_{s_i}^{s_{i+1}} (C'(x) - f'_i(x))^2 dx \right) \quad (4.27)$$

Wypukłość i ograniczenie górne użytej normy w budowaniu wzoru (4.27) oraz liniowe warunki, gwarantują istnienie jednego minimum warunkowego funkcji $Z(\mathbf{v})$.

Natomiast pierwiastek wartości funkcji (4.27),

$$\varepsilon = \sqrt{Z(\mathbf{v})} \quad (4.28)$$

czyli euklidesowa norma w produkcie kartezjańskim przestrzeni Sobolewa nad każdym odcinkiem, będzie wyznaczała błąd aproksymacji ε . Wartość tego błędu pozwoli wyznaczyć dokładność aproksymacji i w konsekwencji górne oszacowanie błędu w normie supremum.

4.4 Proponowana metoda wyznaczania współczynników wielomianów

Dysponując samymi warunkami koniecznymi oraz opcjonalnymi z rozdziału 4.2 nie można wyznaczyć jednoznacznie współczynników wielomianów aproksymujących. Dlatego prezentowana w pracy metoda pozwala wyznaczyć funkcję minimalizującą (4.27), której ekstremum warunkowe jednoznacznie wyznacza wartości współczynników wielomianów aproksymujących.

Zagadnienie to znane jest w teorii optymalizacji jako wyznaczanie ekstremum warunkowego funkcji różniczkowalnej, a jedną z metod jego rozwiązywania jest metoda mnożników Lagrange'a [19].

4.4.1 Postać macierzowa równania

Przedstawione odwzorowanie (4.25) można zapisać w postaci wielomianu stopnia drugiego. Istotnie, wychodząc ze wzoru (4.27)

$$\begin{aligned} Z(\mathbf{v}) = Z(a_0, b_0, c_0, d_0, \dots, a_{N-1}, b_{N-1}, c_{N-1}, d_{N-1}) = \\ \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} (C(x) - (a_i x^3 + b_i x^2 + c_i x + d_i))^2 dx + \\ \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} (C'(x) - (3a_i x^2 + 2b_i x + c_i))^2 dx \end{aligned} \quad (4.29)$$

oraz upraszczając wyrażenia pod całkami

$$\begin{aligned}
& Z(a_0, b_0, c_0, d_0, \dots, a_{N-1}, b_{N-1}, c_{N-1}, d_{N-1}) = \\
& = \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C(x)^2 - 2C(x)(a_i x^3 + b_i x^2 + c_i x + d_i) + (a_i x^3 + b_i x^2 + c_i x + d_i)^2 dx \\
& + \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C'(x)^2 - 2C'(x)(3a_i x^2 + 2b_i x + c_i) + (3a_i x^2 + 2b_i x + c_i)^2 dx = \\
& = \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C(x)^2 dx - 2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C(x)(a_i x^3 + b_i x^2 + c_i x + d_i) dx \\
& + \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} (a_i x^3 + b_i x^2 + c_i x + d_i)^2 dx + \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C'(x)^2 dx \\
& - 2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C'(x)(3a_i x^2 + 2b_i x + c_i) dx + \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} (3a_i x^2 + 2b_i x + c_i)^2 dx
\end{aligned} \tag{4.30}$$

otrzymujemy równanie

$$\begin{aligned}
& Z(a_0, b_0, c_0, d_0, \dots, a_{N-1}, b_{N-1}, c_{N-1}, d_{N-1}) = \\
& \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C(x)^2 dx - 2a_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^3 C(x) dx - 2b_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^2 C(x) dx \\
& - 2c_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x C(x) dx - 2d_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C(x) dx + a_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^6 dx \\
& + 2a_i b_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^5 dx + 2a_i c_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^4 dx + 2a_i d_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^3 dx \\
& + b_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^4 dx + 2b_i c_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^3 dx + 2b_i d_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^2 dx \\
& + c_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x^2 dx + 2c_i d_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} x dx + d_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} dx \\
& + \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C'(x)^2 dx - 2a_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 3x^2 C'(x) dx - 2b_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 2x C'(x) dx \\
& - 2c_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} C'(x) dx + a_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 9x^4 dx + 2a_i b_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 6x^3 dx \\
& + 2a_i c_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 3x^2 dx + b_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 4x^2 dx + 2b_i c_i \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} 2x dx + c_i^2 \sum_{i=0}^{N-1} \int_{s_i}^{s_{i+1}} dx
\end{aligned} \tag{4.31}$$

Analizując to równanie można zauważyć, że poszczególne całki nie są zależne od parametrów występujących w wektorze \mathbf{v} , lecz tylko od węzłów $s_0, s_1, \dots, s_{N-1}, s_N$. Ponieważ prezentowana metoda traktuje te zmienne jako wartości zadane, powyższe całki, określone na odpowiednich przedziałach, można traktować jako wartości stałe

$$\left\{ \begin{array}{l} H_0^i = \int_{s_i}^{s_{i+1}} C(x)^2 dx, \quad i = 0, \dots, N-1 \\ F_0^i = \int_{s_i}^{s_{i+1}} x^3 C(x) dx, \quad i = 0, \dots, N-1 \\ F_1^i = \int_{s_i}^{s_{i+1}} x^2 C(x) dx, \quad i = 0, \dots, N-1 \\ F_2^i = \int_{s_i}^{s_{i+1}} x C(x) dx, \quad i = 0, \dots, N-1 \\ F_3^i = \int_{s_i}^{s_{i+1}} C(x) dx, \quad i = 0, \dots, N-1 \\ G_0^i = \int_{s_i}^{s_{i+1}} x^6 dx, \quad i = 0, \dots, N-1 \\ G_1^i = \int_{s_i}^{s_{i+1}} x^5 dx, \quad i = 0, \dots, N-1 \\ G_2^i = \int_{s_i}^{s_{i+1}} x^4 dx, \quad i = 0, \dots, N-1 \\ G_3^i = \int_{s_i}^{s_{i+1}} x^3 dx, \quad i = 0, \dots, N-1 \\ G_4^i = \int_{s_i}^{s_{i+1}} x^2 dx, \quad i = 0, \dots, N-1 \\ G_5^i = \int_{s_i}^{s_{i+1}} x dx, \quad i = 0, \dots, N-1 \\ G_6^i = \int_{s_i}^{s_{i+1}} dx, \quad i = 0, \dots, N-1 \\ H_0^i = \int_{s_i}^{s_{i+1}} C(x)^2 dx, \quad i = 0, \dots, N-1 \\ F_0^i = \int_{s_i}^{s_{i+1}} 3x^2 C'(x) dx, \quad i = 0, \dots, N-1 \\ F_1^i = \int_{s_i}^{s_{i+1}} 2x C'(x) dx, \quad i = 0, \dots, N-1 \\ F_2^i = \int_{s_i}^{s_{i+1}} C'(x) dx, \quad i = 0, \dots, N-1 \\ G_0^i = \int_{s_i}^{s_{i+1}} 9x^4 dx, \quad i = 0, \dots, N-1 \\ G_1^i = \int_{s_i}^{s_{i+1}} 6x^3 dx, \quad i = 0, \dots, N-1 \\ G_2^i = \int_{s_i}^{s_{i+1}} 3x^2 dx, \quad i = 0, \dots, N-1 \\ G_3^i = \int_{s_i}^{s_{i+1}} 4x^2 dx, \quad i = 0, \dots, N-1 \\ G_4^i = \int_{s_i}^{s_{i+1}} 2x dx, \quad i = 0, \dots, N-1 \\ G_5^i = \int_{s_i}^{s_{i+1}} dx, \quad i = 0, \dots, N-1 \end{array} \right. \quad (4.32)$$

W ten sposób wzór (4.27) został przekształcony do postaci funkcji drugiego stopnia wielu zmiennych (4.33)

$$\begin{aligned}
& Z(a_0, b_0, c_0, d_0, \dots, a_{N-1}, b_{N-1}, c_{N-1}, d_{N-1}) = \\
& = H_0^0 - 2a_0F_0^0 - 2b_0F_1^0 - 2c_0F_2^0 - 2d_0F_3^0 + a_0^2G_0^0 + \\
& + 2a_0b_0G_1^0 + 2a_0c_0G_2^0 + 2a_0d_0G_3^0 + b_0^2G_2^0 + 2b_0c_0G_4^0 + \\
& + 2b_0d_0G_5^0 + c_0^2G_4^0 + 2c_0d_0G_5^0 + d_0^2G_6^0 + \\
& + H_0'^0 - 2a_0F_0'^0 - 2b_0F_1'^0 - 2c_0F_2'^0 + a_0^2G_0'^0 + \\
& + 2a_0b_0G_1'^0 + 2a_0c_0G_2'^0 + b_0^2G_3'^0 + 2b_0c_0G_4'^0 + c_0^2G_5'^0 + \\
& + \dots + \\
& + H_0^{N-1} - 2a_{N-1}F_0^{N-1} - 2b_{N-1}F_1^{N-1} - 2c_{N-1}F_2^{N-1} + \\
& - 2d_{N-1}F_3^{N-1} + a_{N-1}^2G_0^{N-1} + 2a_{N-1}b_{N-1}G_1^{N-1} + \\
& + 2a_{N-1}c_{N-1}G_2^{N-1} + 2a_{N-1}d_{N-1}G_3^{N-1} + b_{N-1}^2G_2^{N-1} + \\
& + 2b_{N-1}c_{N-1}G_4^{N-1} + 2b_{N-1}d_{N-1}G_5^{N-1} + c_{N-1}^2G_4^{N-1} \\
& + 2c_{N-1}d_{N-1}G_5^{N-1} + d_{N-1}^2G_6^{N-1} \\
& + H_0'^{N-1} - 2a_{N-1}F_0'^{N-1} - 2b_{N-1}F_1'^{N-1} - 2c_{N-1}F_2'^{N-1} + \\
& + a_{N-1}^2G_0'^{N-1} + 2a_{N-1}b_{N-1}G_1'^{N-1} + 2a_{N-1}c_{N-1}G_3'^{N-1} + \\
& + b_{N-1}^2G_2'^{N-1} + 2b_{N-1}c_{N-1}G_4'^{N-1} + c_{N-1}^2G_5'^{N-1}
\end{aligned} \tag{4.33}$$

Wzór (4.33) można przedstawić w postaci macierzowej. Ze współczynników stojących przy drugiej potędze niewiadomych $a_0, \dots, a_{N-1}, b_0, \dots, b_{N-1}, c_0, \dots, c_{N-1}, d_0, \dots, d_{N-1}$ utworzone zostaną macierze G oraz G'

$$G = \begin{bmatrix} G_0^0 & G_1^0 & G_2^0 & G_3^0 & \dots & 0 & 0 & 0 & 0 \\ G_1^0 & G_2^0 & G_4^0 & G_5^0 & \dots & 0 & 0 & 0 & 0 \\ G_2^0 & G_4^0 & G_4^0 & G_5^0 & \dots & 0 & 0 & 0 & 0 \\ G_3^0 & G_5^0 & G_5^0 & G_6^0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & G_0^{N-1} & G_1^{N-1} & G_2^{N-1} & G_3^{N-1} \\ 0 & 0 & 0 & 0 & \dots & G_1^{N-1} & G_2^{N-1} & G_4^{N-1} & G_5^{N-1} \\ 0 & 0 & 0 & 0 & \dots & G_2^{N-1} & G_4^{N-1} & G_4^{N-1} & G_5^{N-1} \\ 0 & 0 & 0 & 0 & \dots & G_3^{N-1} & G_5^{N-1} & G_5^{N-1} & G_6^{N-1} \end{bmatrix} \tag{4.34}$$

$$G' = \begin{bmatrix} G_0'^0 & G_1'^0 & G_2'^0 & 0 & \dots & 0 & 0 & 0 & 0 \\ G_1'^0 & G_3'^0 & G_4'^0 & 0 & \dots & 0 & 0 & 0 & 0 \\ G_2'^0 & G_4'^0 & G_5'^0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & G_0'^{N-1} & G_1'^{N-1} & G_2'^{N-1} & 0 \\ 0 & 0 & 0 & 0 & \dots & G_1'^{N-1} & G_3'^{N-1} & G_4'^{N-1} & 0 \\ 0 & 0 & 0 & 0 & \dots & G_2'^{N-1} & G_4'^{N-1} & G_5'^{N-1} & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.35}$$

natomiast ze współczynników przy wyrazach liniowych można utworzyć wektory \mathbf{f} oraz \mathbf{f}' .

$$\mathbf{f} = \begin{bmatrix} F_0^0 & F_1^0 & F_2^0 & F_3^0 & \dots & F_0^{N-1} & F_1^{N-1} & F_2^{N-1} & F_3^{N-1} \end{bmatrix} \quad (4.36)$$

$$\mathbf{f}' = \begin{bmatrix} F_0'^0 & F_1'^0 & F_2'^0 & 0 & \dots & F_0'^{N-1} & F_1'^{N-1} & F_2'^{N-1} & 0 \end{bmatrix} \quad (4.37)$$

Dla pozostałych wyrazów

$$h = H_0^0 + \dots + H_0^{N-1} \quad (4.38)$$

$$h' = H_0'^0 + \dots + H_0'^{N-1} \quad (4.39)$$

Zatem równanie (4.33) można zapisać w postaci

$$Z(\mathbf{v}) = \mathbf{v}(\mathbf{G} + \mathbf{G}')\mathbf{v}^T - 2\mathbf{v}(\mathbf{f} + \mathbf{f}')^T + (h + h') \quad (4.40)$$

$$Z(\mathbf{v}) = \underset{4N \times 1}{\mathbf{v}} \times \underset{4N \times 4N}{(\mathbf{G} + \mathbf{G}')} \times \underset{1 \times 4N}{\mathbf{v}^T} - 2 * \underset{4N \times 1}{\mathbf{v}} \times \underset{1 \times 4N}{(\mathbf{f} + \mathbf{f}')^T} + (h + h') \quad (4.41)$$

Wzór funkcji (4.41) jest poprawny tylko dla funkcji wejściowych $C(x)$ z rodziny Ω_+ . dla których jest możliwość wyznaczenia całek F_0^i oraz H_0^i dla $i = 0, \dots, N-1$. Dla funkcji rodziny Ω_1 wzór przedstawia się następująco:

$$Z(\mathbf{v}) = \underset{4N \times 1}{\mathbf{v}} \times \underset{4N \times 4N}{\mathbf{G}} \times \underset{1 \times 4N}{\mathbf{v}^T} - 2 * \underset{4N \times 1}{\mathbf{v}} \times \underset{1 \times 4N}{\mathbf{f}^T} + h \quad (4.42)$$

Postać równania zależy tylko od liczby N wielomianów użytych do aproksymacji (od liczby punktów podziału aproksymowanej krzywej). Dodatkowo, macierze \mathbf{G} oraz \mathbf{G}' zależą tylko od węzłów aproksymacji, a nie od funkcji wejściowej $C(x)$, w przeciwieństwie do \mathbf{f} oraz \mathbf{f}' . Ponadto wyrazy wolne mogą być wyznaczone jako całki w granicach węzłów zewnętrznych (nie jest konieczne liczenie ich jako sumy całek w przedziałach wielomianów aproksymujących).

4.4.2 Warunki ciągłości w postaci macierzy rozszerzonej

Warunki konieczne oraz opcjonalne opisane w rozdziałach 4.1 oraz 4.2.2 w postaci macierzy rozszerzonej przedstawić można następująco (pogrupowane tak by warunki różnych klas gładkości w węzłach były sąsiadujące)

- dla warunku (4.3)

$$\left[\begin{array}{cccc|cccc} s_0^3 & s_0^2 & s_0 & 1 & \dots & 0 & 0 & 0 & 0 & C(s_0) \\ 0 & 0 & 0 & 0 & \dots & s_N^3 & s_N^2 & s_N & 1 & C(s_{N-1}) \end{array} \right] \quad (4.43)$$

- dla warunku (4.9)

$$\left[\begin{array}{cccc|cccc} 3s_0^2 & 2s_0 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & C'(s_0) \\ 0 & 0 & 0 & 0 & \dots & 3s_N^2 & 2s_N & 1 & 0 & C'(s_N) \end{array} \right] \quad (4.44)$$

- dla warunku (4.10)

$$\left[\begin{array}{cccccc|cccc} 6s_0 & 2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & C'''(s_0) \\ 0 & 0 & 0 & 0 & \dots & 6s_N & 2 & 0 & 0 & C'''(s_N) \end{array} \right] \quad (4.45)$$

- dla warunku (4.5)

$$\left[\begin{array}{cccccccc|cccc} s_1^3 & s_1^2 & s_1 & 1 & -s_1^3 & -s_1^2 & -s_1 & -1 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ s_2^3 & s_2^2 & s_2 & 1 & -s_2^3 & -s_2^2 & -s_2 & -1 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & s_{N-2}^3 & s_{N-2}^2 & s_{N-2} & 1 & -s_{N-2}^3 & -s_{N-2}^2 & -s_{N-2} & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & s_{N-1}^3 & s_{N-1}^2 & s_{N-1} & 1 & -s_{N-1}^3 & -s_{N-1}^2 & -s_{N-1} & -1 \end{array} \right] \quad (4.46)$$

- dla warunku (4.7)

$$\left[\begin{array}{cccccccc|cccc} 3s_1^2 & 2s_1 & 1 & 0 & -3s_1^2 & -2s_1 & -1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 3s_2^2 & 2s_2 & 1 & 0 & -3s_2^2 & -2s_2 & -1 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 3s_{N-2}^2 & 2s_{N-2} & 1 & 0 & -3s_{N-2}^2 & -2s_{N-2} & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 3s_{N-1}^2 & 2s_{N-1} & 1 & 0 & -3s_{N-1}^2 & -2s_{N-1} & -1 & 0 \end{array} \right] \quad (4.47)$$

- dla warunku (4.8)

$$\left[\begin{array}{cccccc|cccc} 6s_1 & 2 & 0 & 0 & -6s_1 & -2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 6s_2 & 2 & 0 & 0 & -6s_2 & -2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 6s_{N-2} & 2 & 0 & 0 & -6s_{N-2} & -2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \dots & 6s_{N-1} & 2 & 0 & 0 & -6s_{N-1} & -2 & 0 \end{array} \right] \quad (4.48)$$

- dla warunku (4.6)

$$\left[\begin{array}{cccccc|cccc} s_1^3 & s_1^2 & s_1 & 1 & \dots & 0 & 0 & 0 & 0 & C(s_1) \\ s_2^3 & s_2^2 & s_2 & 1 & \dots & 0 & 0 & 0 & 0 & C(s_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & s_{N-2}^3 & s_{N-2}^2 & s_{N-2} & 1 & C(s_{N-2}) \\ 0 & 0 & 0 & 0 & \dots & s_{N-1}^3 & s_{N-1}^2 & s_{N-1} & 1 & C(s_{N-1}) \end{array} \right] \quad (4.49)$$

- dla warunku (4.11)

$$\left[\begin{array}{cccccccc|c} 3s_1^2 & 2s_1 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & C'(s_1) \\ 3s_2^2 & 2s_2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 & C'(s_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 3s_{N-2}^2 & 2s_{N-2} & 1 & 0 & C'(s_{N-2}) \\ 0 & 0 & 0 & 0 & \dots & 3s_{N-1}^2 & 2s_{N-1} & 1 & 0 & C'(s_{N-1}) \end{array} \right] \quad (4.50)$$

- dla warunku (4.12)

$$\left[\begin{array}{cccccccc|c} 6s_1 & 2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & C''(s_1) \\ 6s_2 & 2 & 0 & 0 & \dots & 0 & 0 & 0 & 0 & C''(s_2) \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 6s_{N-2} & 2 & 0 & 0 & C''(s_{N-2}) \\ 0 & 0 & 0 & 0 & \dots & 6s_{N-1} & 2 & 0 & 0 & C''(s_{N-1}) \end{array} \right] \quad (4.51)$$

4.5 Ekstrema warunkowe funkcji różniczkowalnych

Zastosowanie tylko warunków koniecznych oraz opcjonalnych (w większości przypadków) nie daje możliwości jednoznacznego wyznaczenia współczynników wielomianów aproksymujących, gdyż liczba warunków koniecznych oraz opcjonalnych jest mniejsza niż liczba poszukiwanych współczynników. Dlatego proponowana metoda znajduje minimum funkcji celu (4.41) (dla przypadków funkcji brzegowych (4.42)) leżące na powierzchni opisanej przez warunki konieczne oraz opcjonalne. Taki problem w teorii optymalizacji nosi nazwę ekstremum warunkowego funkcji różniczkowalnej [19]. Takie podejście pozwoli jednoznacznie wyznaczyć współczynniki poszukiwanych wielomianów i wielomiany te będą minimalizować odległość od funkcji wejściowej, w normie $W^{1,2}$.

4.5.1 Metoda mnożników Lagrange'a

Metoda dotyczy funkcji różniczkowalnych $f : \mathbb{R}^n \Rightarrow \mathbb{R}$, dla których szukamy ekstremów na podzbiorze w przestrzeni \mathbb{R}^n , zadanym przez zdefiniowany dalej układ równań $G(x) = 0$, czyli poziomicy przekształcenia $G : \mathbb{R}^n \Rightarrow \mathbb{R}^m$, które jest przekształceniem różniczkowalnym. Zakładamy, że w każdym punkcie zerowej poziomicy przekształcenia G jego macierz Jacobiego jest maksymalnego rzędu. Jeśli G jest klasy C^1 , to warunek ten oznacza, że poziomica ta jest rozmaitością różniczkowalną klasy C^1 . Układ równań Lagrange'a przybiera następującą postać:

$$\begin{cases} f'(x) = \Lambda \circ G'(x) \\ G(x) = 0 \end{cases} \quad (4.52)$$

gdzie Λ jest pewnym funkcjonałem liniowym na \mathbb{R}^m oraz G' jest macierzą Jacobiego przekształcenia G . Zbiór tych punktów dla których $G(x) = 0$, jest niczym innym jak poziomicą zerową przekształcenia G .

Jak wiadomo każdy funkcjonał Λ określony na przestrzeni euklidesowej \mathbb{R}^m reprezentowany jest przez pewien jednoznacznie wyznaczony przez niego wektor $[\lambda_1, \dots, \lambda_m] \in \mathbb{R}^m$ (twierdzenie Riesz [31],[19]).

Powyższy układ równań jest więc układem $m + n$ równań skalarnych:

$$\begin{cases} \frac{\partial f(x)}{\partial x_j} = \sum_{i=1}^m \lambda_i \frac{\partial G_i(x)}{\partial x_j}, & j = 1, \dots, n \\ G_k(x_1, \dots, x_n) = 0, & k = 1, \dots, m \end{cases} \quad (4.53)$$

gdzie $x = (x_1, \dots, x_n)$ o $n + m$, $\frac{\partial G_i(x)}{\partial x_j}$ są pochodnymi cząstkowymi.

Wszystkie punkty, w których funkcja może przyjmować ekstrema warunkowe, należą do zbioru rozwiązań tego układu równań. Liczby λ_i spełniają tylko rolę pomocniczą i nazywane są często mnożnikami Lagrange'a [19]. Powyższe warunki są warunkami koniecznymi dla istnienia ekstremów warunkowych. Punkty w \mathbb{R}^{m+n} spełniające ten układ nazywamy warunkowymi punktami krytycznymi. W przypadku dwukrotnej różniczkowalności funkcji f , o tym czy punkt krytyczny x_0 jest ekstremum warunkowym pomaga rozstrzygnąć określoność formy kwadratowej wyznaczonej przez macierz

$$H(x) = f''(x) - \Lambda \circ G''(x) \quad (4.54)$$

gdzie f'' jest macierzą drugich pochodnych cząstkowych w punkcie x_0 funkcji f (Hesjanem funkcji). W tym przypadku określoność tej formy badamy wyłącznie na wektorach h należących do jądra przekształcenia G' , co sprowadza się do badania określoności formy kwadratowej na pewnej podprzestrzeni liniowej w \mathbb{R}^m .

Po znalezieniu punktów spełniających warunek konieczny dla ekstremum (punktów stacjonarnych), możemy odwołać się do warunku dostatecznego istnienia ekstremum, tj. zbadać dodatnią (ujemną) określoność co sprowadza się do badania formy kwadratowej

$$\sum_{i,j=1}^n \left(\frac{\partial^2 f(x_0)}{\partial x_i \partial x_j} - \sum_{k=1}^m \lambda_k \frac{\partial^2 G_k(x_0)}{\partial x_i \partial x_j} \right) h_i h_j \quad (4.55)$$

gdzie x_0 jest punktem stacjonarnym, oraz $h = (h_1, \dots, h_n)$,

$$h \in X_1 = \ker G'(x_0) \quad (4.56)$$

$\ker G'(x_0)$ oznacza jądro przekształcenia liniowego $G'(x_0)$, czyli podprzestrzeń liniową tych wektorów, które macierz $G'(x_0)$ przekształca w wektor zerowy.

Warunek $h \in X_1$ jest równoważny równaniu

$$G'(x)h = 0 \quad (4.57)$$

które w postaci macierzowej przybiera formę

$$\sum_{i=1}^n \frac{\partial G_k(x)}{\partial x_i} h_i = 0, \quad k = 1, 2, \dots, m \quad (4.58)$$

W prezentowanej metodzie problem znajdowania ekstremum warunkowego przy użyciu metody mnożników Lagrange'a sprowadza się do rozwiązania pewnego układu równań liniowych na podstawie wzoru (4.53). I można przedstawić jako równanie macierzowe

$$W\mathbf{x} = \mathbf{y} \quad (4.59)$$

gdzie

$$W = \left[\begin{array}{c|c} \mathbf{M} & -\mathbf{A}^T \\ \hline \mathbf{A} & 0 \end{array} \right] \quad (4.60)$$

gdzie \mathbf{M} to macierz $\mathbf{M} = \mathbf{G} + \mathbf{G}'$ z równania (4.41) dla rodziny funkcji Ω_+ oraz $\mathbf{M} = \mathbf{G}$ dla rodzin funkcji Ω_1 , oraz macierz \mathbf{A} każdorazowo konstruowana jest z prawych stron macierzy rozszerzonych, wymienionych w rozdziale 4.4.2, w zależności od wybranych warunków gładkości w węzłach.

Wektor

$$\mathbf{x} = \left[\begin{array}{c} \mathbf{v} \\ \lambda \end{array} \right] \quad (4.61)$$

składa się z poszukiwanych współczynników wielomianów (tworzą one wektor \mathbf{v}) oraz mnożników Lagrange'a (tworzą one wektor λ).

Prawa strona równania (4.59), czyli wektor \mathbf{y} , opisany jest wzorem

$$\mathbf{y} = \left[\begin{array}{c} \mathbf{k} \\ \mathbf{D} \end{array} \right] \quad (4.62)$$

gdzie $\mathbf{k} = \mathbf{f} + \mathbf{f}'$ z równania (4.41) dla rodziny funkcji Ω_+ oraz $\mathbf{k} = \mathbf{f}$ dla rodzin funkcji Ω_1 . Wektor \mathbf{D} składa się z lewych stron macierzy rozszerzonych warunków koniecznych oraz warunków opcjonalnych opisanych w rozdziale 4.4.2.

4.6 Możliwe rozszerzenie metody o aproksymacje wielomianami wyższych stopni

Chociaż w tezie prezentowanej pracy wskazano wielomian co najwyżej trzeciego stopnia jako funkcję aproksymującą krzywą Béziera trzeciego stopnia, to prezentowana metoda jest na tyle elastyczna, że może być rozszerzona o aproksymację trajektorii $C(x)$ przy użyciu innych funkcji, w tym wielomianów wyższych stopni.

Aproksymując funkcje wejściową wielomianami k -go stopnia,

$$C(x) \approx \sum_{i=0}^{N-1} f_i(x) \quad (4.63)$$

takimi że

$$f_i(x) = \sum_{j=0}^k a_j^i x^j \quad x \in [s_i, s_{i+1}] \quad i = 0, 1, \dots, N-2, N-1 \quad (4.64)$$

funkcja minimalizująca

$$Z(\mathbf{v}) : \mathbb{R}^{kN} \Rightarrow \mathbb{R} \quad (4.65)$$

gdzie N to liczba wielomianów co najwyżej k -go stopnia, a wektor v zbudowany jest ze współczynników wielomianów aproksymujących.

$$\mathbf{v} = [a_k^0, a_{k-1}^0, \dots, a_1^0, a_0^0, \dots, a_k^N, a_{k-1}^N, \dots, a_1^N, a_0^N] \quad (4.66)$$

Postać funkcji $Z(\mathbf{v})$ zbudowana jest w tym przypadku z wykorzystaniem kwadratu normy $W^{1,2}$

$$Z(\mathbf{v}) = \sum_{i=0}^{N-1} \left(\int_{s_i}^{s_{i+1}} (C(x) - f_i(x))^2 dx + \int_{s_i}^{s_{i+1}} (C'(x) - f_i'(x))^2 dx \right) \quad (4.67)$$

Postępując następnie w analogiczny sposób jak w rozdziale 4.4.1 wyznaczyć można funkcję minimalizującą w postaci równania macierzowego. Kolejnym krokiem byłoby wyznaczenie warunków ciągłości z rozdziału 4.2 (wielomiany są funkcjami należącymi do każdej klasy C^k , możliwe jest wprowadzanie każdego warunku ciągłości w węzłach aproksymacji) i przedstawienie ich w postaci macierzy. Do wyznaczenia poszukiwanych współczynników wielomianów można zastosować tę samą metodę jak dla badanych wielomianów trzeciego stopnia, czyli metodę mnożników Lagrange'a.

Z praktycznego punktu widzenia stopień wielomianu powinien być ograniczony, gdyż każde jego zwiększenie narzuca zwiększenie zapotrzebowania pamięciowego na zapis jego współczynników oraz konieczność wyznaczenia całek (4.32), a ich postać implikuje zwiększającą się, wraz ze stopniem wielomianu, koszt obliczeń.

4.7 Problem wyznaczenia optymalnych węzłów

Metoda proponowana w pracy przedstawiając wzór (4.27) nie dostarcza metody wyznaczenia węzłów wewnętrznych $s_1, s_2, \dots, s_{N-2}, s_{N-1}$. Ten sam problem istniał w interpolacji funkcjami sklejanymi. Najlepszym sposobem byłoby uwzględnienie tych zmiennych w poszukiwaniu minimum funkcji (4.27). Niestety znalezienie tego minimum nie jest możliwe znanymi metodami algebraicznymi (mamy tu do czynienia z układem równań nieliniowych). Jednym ze sposobów

mogłoby być rozwiązanie tego równania metodami numerycznymi, co daleko wykracza poza zakres tej pracy.

Problem ten można rozwiązać wykorzystując podział równomierny wejściowego przedziału x_0, x_1 wzdłuż krzywej $C(x)$ lub podział nierównomierny z wykorzystaniem metody siłowej.

Do znalezienia najlepszych podziałów można wykorzystać metodę **siłową** (ang. *brute force*). Wejściowy przedział $[x_0, x_1]$ (3.7) dzielony jest przy pomocy zbioru $K - 1$ -elementowego równomiernie położonych, potencjalnych węzłów $u_0, u_1, \dots, u_{K-1}, u_K$ (przy dodatkowym założeniu $u_i < u_{i+1}$ $i = 1, 2, \dots, K - 2, K - 1$). W ten sam sposób dzielenie odbywa się wzdłuż wejściowej krzywej $C(x)$, gdzie

$$\begin{cases} u_0 = x_0 \\ u_K = x_1 \end{cases} \quad (4.68)$$

Następnie wyznaczany jest błąd aproksymacji (4.27) dla każdej kombinacji bez powtórzeń elementów z tego zbioru. Kombinacja potencjalnych węzłów, dla których wartość błędu jest najmniejsza, wyznacza poszukiwane węzły $s_1, s_2, \dots, s_{N-2}, s_{N-1}$ wielomianów aproksymujących. Liczba testowanych kombinacji pod podziałów do wyznaczenia najlepszego rośnie bardzo szybko (złożoność obliczeniowa rzędu $N!$).

Złożoność obliczeniową metody siłowej można poprawiać stosując metody heurystyczne, na przykład można wykorzystać następujący algorytm zachłanny 4 dla aproksymacji wykorzystującej dwa wielomiany co najwyżej trzeciego stopnia. Metoda heurystyczna polega na podziale przedziału wejściowego (podobnie jak w metodzie siłowej) na K -elementowego równomiernie położonych, potencjalnych węzłów $u_0, u_1, \dots, u_{K-1}, u_K$ gdzie $K < 12$. Po wyznaczeniu najlepszego podziału przedziału wejściowego, znowu wyznacza się K -elementowy równomiernie położone, potencjalne węzły ale tym razem w okolicy najlepszego przedziału z iteracji poprzedzającej (listing tego algorytmu przedstawia Algorytm 4). Tę operację powtarza się, aż uzyska się błąd aproksymacji mniejszy niż akceptowany lub też po wykonaniu z góry ustalonej liczby iteracji.

Algorytm 4 Listing przykładowej optymalizacji przy użyciu metody heurystycznej.

```
function HEURISTIC-FUNCTION( $t_b, t_e, K, errorLimit$ )  
  for  $i \leftarrow 1$  to  $K$  do  
     $x = \frac{i*(t_e-t_b)/K-t_b}{t_e-t_b}$   
     $\varepsilon = \text{wyznacz-bład}(x)$   
     $best = \text{sprawdz-czy-przedział-jest-lepszy-od-poprzednich}$   
  end for  
  if  $\varepsilon > errorLimit$  then  
     $\tau_b = \frac{(best-1)*(t_e-t_b)/K-t_b}{t_e-t_b}$   
     $\tau_e = \frac{(best+1)*(t_e-t_b)/K-t_b}{t_e-t_b}$   
    HEURISTIC-FUNCTION( $\tau_b, \tau_e, K$ )  
  else  
    return  $\varepsilon, best$   
  end if  
end function
```

Taki algorytm nie jest analizowany w prezentowanej pracy, ponieważ w niewielkim stopniu wpływa na znalezienie węzłów z najmniejszym błędem aproksymacji (lub różnice w błędach aproksymacji są poniżej wartości 0.00001), ma jedynie duży wpływ na zmniejszenie złożoności obliczeniowej (w porównaniu z prezentowaną metodą siłową), co przekłada się na czas potrzebny do wyznaczenia najlepszych węzłów.

4.8 Wnioski

Zaproponowana metoda, czyli wyznaczenie funkcji celu (4.27) której minimalizacja przy dodatkowych warunkach koniecznych oraz opcjonalnych spełnia założenia tezy, dając metodę znajdowania współczynników wielomianów, minimalizując bład mierzony normy w przestrzeni Sobolewa $W^{1,2}$. Użyta norma przestrzeni Sobolewa $W^{1,2}$ umożliwia górne oszacowanie błędu aproksymacji w normie supremum.

Zaproponowana metoda wymagała dodatkowych testów. Testy te zostały opisane w rozdziale 6, gdzie sprawdzono, jak narzucone warunki np. liczba wielomianów aproksymujących czy też warunki ciągłości wielomianów w węzłach, wpływają na bład aproksymujący.

Rozdział 5

Koncepcja sprawdzania poprawności rozwiązania

5.1 Wstępne informacje potrzebne do analizy proponowanej metody

Wartości współrzędnych punktów kontrolnych krzywej Béziera (3.3) nie są niczym ograniczone w zastosowaniach modelowania geometrycznego. Jednakże w zastosowaniu krzywej Béziera, jako funkcji interpolacyjnej w metodzie klatek kluczowych, współrzędne punktów kontrolnych ograniczone są przez warunki (3.13) lub (3.14). Analiza wszystkich możliwych przypadków jest praktycznie niemożliwa. Dlatego też zaproponowane rozwiązanie pozwala wyznaczyć postać znormalizowaną krzywej Béziera (dokładny opis tej postaci przedstawiony jest w rozdziale 5.1.1), aby przy użyciu tej postaci ułatwić testy oraz analizę prezentowanej metody.

5.1.1 Postać znormalizowana krzywej Béziera trzeciego stopnia

Jedną z podstawowych właściwości krzywej Béziera jest „*Niezmienniczość reprezentacji krzywej Béziera ze względu na przekształcenia afiniczne*”[16]. Dowolną krzywą Béziera trzeciego stopnia (o punktach kontrolnych P_0, P_1, P_2, P_3), która spełnia warunek (3.13) lub (3.14), można przekształcić tak, aby punkty kontrolne P'_0 oraz P'_3 znormalizowanej krzywej spełniały następujący warunek:

$$\begin{cases} P'_0 = (0, 0) \\ P'_3 = (1, 1) \end{cases} \quad (5.1)$$

Pozostałe punkty kontrolne (P'_1 oraz P'_2) wyznaczone są przy użyciu następującego przekształcenia:

- dla $Y_{P_0} \neq Y_{P_3}$

$$\begin{cases} P'_0 = (0, 0) \\ P'_1 = \left(\frac{X_{P_1} - X_{P_0}}{X_{P_3} - X_{P_0}}, \frac{Y_{P_1} - Y_{P_0}}{Y_{P_3} - Y_{P_0}} \right) \\ P'_2 = \left(\frac{X_{P_2} - X_{P_0}}{X_{P_3} - X_{P_0}}, \frac{Y_{P_2} - Y_{P_0}}{Y_{P_3} - Y_{P_0}} \right) \\ P'_3 = (1, 1) \end{cases} \quad (5.2)$$

- dla $Y_{P_0} = Y_{P_3}$

$$\begin{cases} P'_0 = (0, 0) \\ P'_1 = \left(\frac{X_{P_1} - X_{P_0}}{X_{P_3} - X_{P_0}}, \frac{X_{P_1} - X_{P_0}}{X_{P_3} - X_{P_0}} + Y_{P_1} \right) \\ P'_2 = \left(\frac{X_{P_2} - X_{P_0}}{X_{P_3} - X_{P_0}}, \frac{X_{P_2} - X_{P_0}}{X_{P_3} - X_{P_0}} + Y_{P_2} \right) \\ P'_3 = (1, 1) \end{cases} \quad (5.3)$$

Krzywa Béziera trzeciego stopnia, która spełnia warunek (5.1) (w dalszej części pracy określana jako **znormalizowana krzywa Béziera trzeciego stopnia**), ograniczy zbiór wszystkich możliwych trajektorii oraz ułatwi analizę proponowanej metody.

5.1.2 Ograniczenia dla postaci znormalizowanej

Narzucając ograniczenia opisane w (5.1) na punkty kontrolne P'_0 oraz P'_3 wymusza się jednocześnie ograniczania współczynników wielomianów wewnętrznych oraz zewnętrznych (w parametryzacji (3.6)), które mają następującą postać:

$$\begin{cases} g(0) = 0 \\ g(1) = 1 \end{cases} \quad (5.4)$$

oraz

$$\begin{cases} f(0) = 0 \\ f(1) = 1 \end{cases} \quad (5.5)$$

co upraszcza postać wielomianów g oraz f do

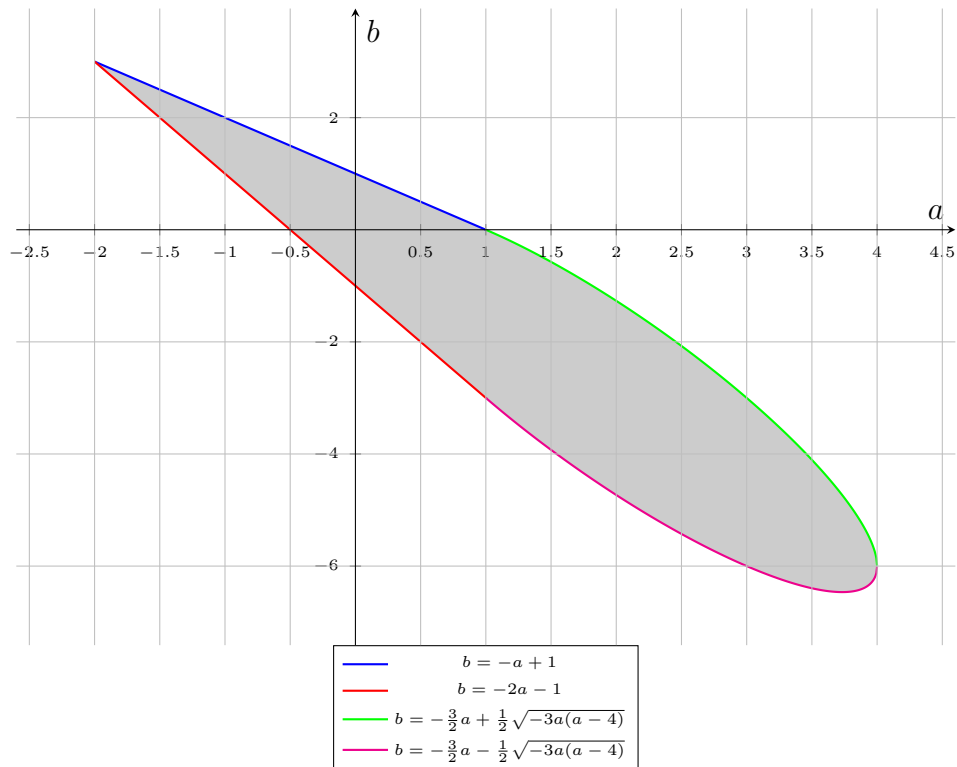
$$g = a_0x^3 + b_0x^2 + (1 - a_0 - b_0)x \quad (5.6)$$

oraz

$$f = a_1x^3 + b_1x^2 + (1 - a_1 - b_1)x \quad (5.7)$$

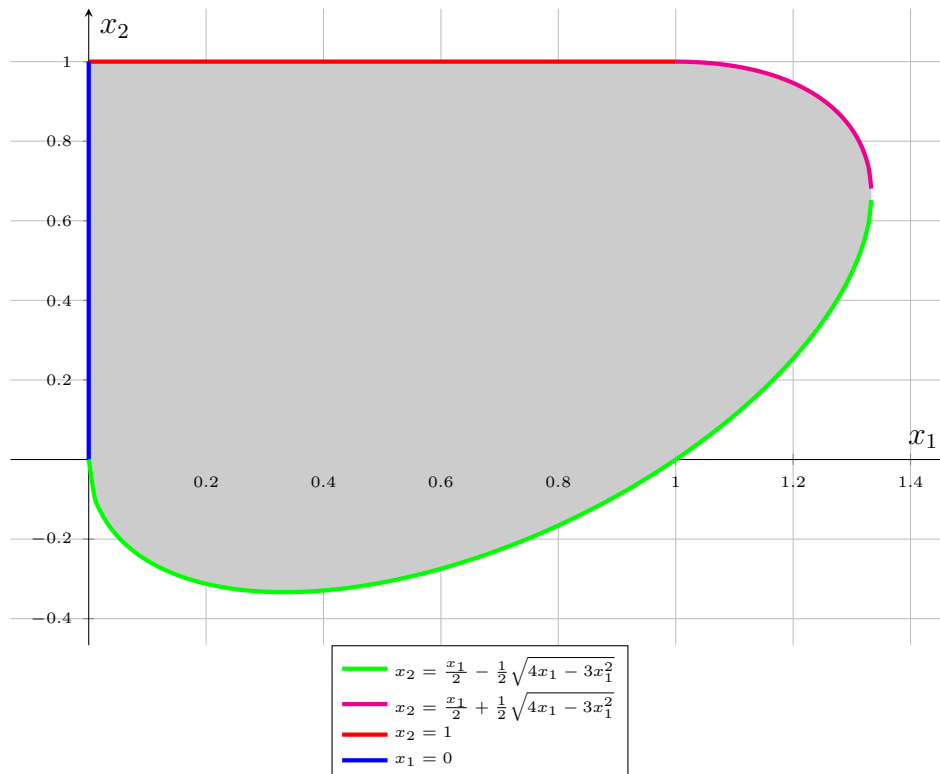
Normalizacja krzywej Béziera trzeciego stopnia spowodowała, że wielomian zewnętrzny f zależny jest tylko od dwóch współczynników tak samo jak wielomian wewnętrzny g .

Znormalizowana krzywa Béziera trzeciego stopnia musi także spełniać warunek (3.13) lub (3.14), by móc być użytą jako funkcja interpolacyjna w animacji komputerowej. Wykres 5.1 przedstawia obszar wartości, jakie mogą przyjmować współczynniki a oraz b wielomianu wewnętrznego (5.6), aby spełniały warunek (3.13) lub (3.14).



Rysunek 5.1: Obszar możliwych wartości współczynników a i b , dla którego wielomian wewnętrzny g spełnia warunek(5.6).

Możliwe jest także przedstawienie wykresu współrzędnych x_1, x_2 punktów kontrolnych P'_1 oraz P'_2 znormalizowanej krzywej Béziera, dla których spełniają one warunek (3.13) lub (3.14) - rysunek 5.2.



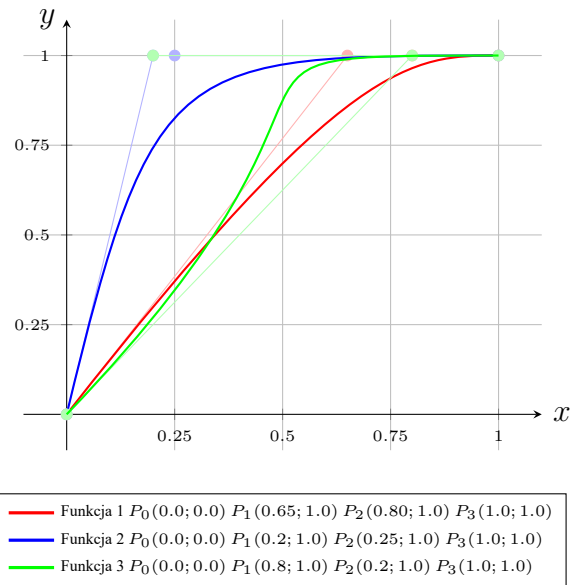
Rysunek 5.2: Obszar możliwych wartości współrzędnych x punktów kontrolnych P_1^l oraz P_2^l znormalizowanej krzywej Béziera.

Warto wspomnieć, że funkcje z rodziny Ω_1 (opisanej w rozdziale 3.1) występują, gdy współczynniki a oraz b określają punkt położony na brzegach obszarów przedstawionych wykresów 5.1 oraz 5.2.

5.1.3 Krzywe Béziera poddane testom

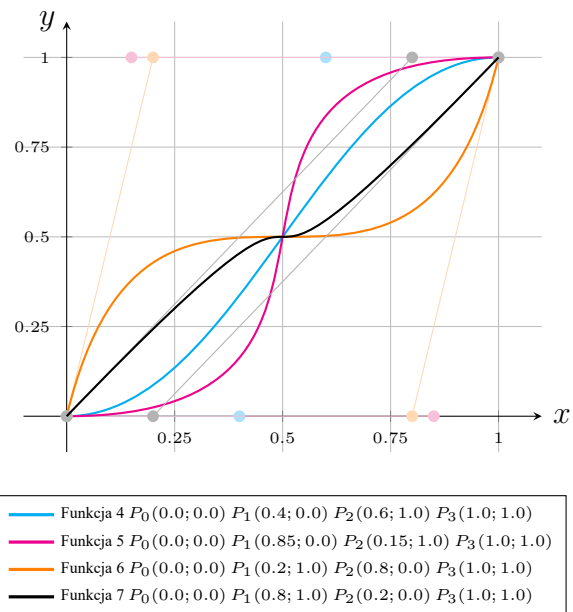
Wszystkie testy zostaną przeprowadzone na zestawie dziesięciu przykładowych, znormalizowanych krzywych Béziera trzeciego stopnia (w dalszej części pracy nazywane **krzywymi testowymi**). Dodatkowo, współrzędne punktów kontrolnych krzywych testowych zostały tak dobrane, aby wyglądem przypominały krzywe zaprezentowane przez Pennera w [27]; zostały one także podzielone na trzy grupy:

- Grupa I. Krzywe podobne do typu *Ease-Out* - rysunek 5.3



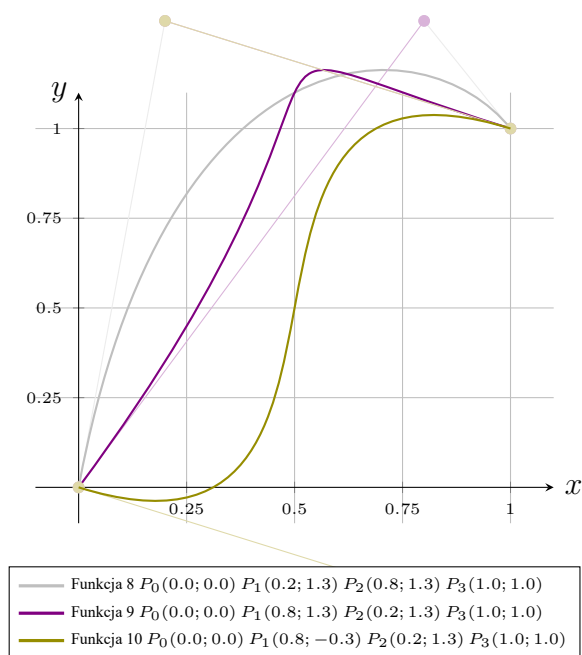
Rysunek 5.3: Wykresy testowych krzywych Béziera z grupy I.

- Grupa II. Krzywe podobne do typu *Ease-InOut* oraz *Ease-OutIn* - rysunek 5.4



Rysunek 5.4: Wykresy testowych krzywych Béziera z grupy II.

- Grupa III. Krzywe podobne do typu *Ease OutBack* oraz *Ease InOutBack* - rysunek 5.5



Rysunek 5.5: Wykresy testowych krzywych Béziera z grupy III.

W prezentowanych powyżej grupach brakuje krzywych podobnych do *Ease In*, gdyż ta grupa jest symetryczna względem grupy *Ease Out*, a wyniki odpowiednich błędów aproksymacji mają podobny charakter.

Warto zaznaczyć że zaprezentowane krzywe testowe nie są krzywymi Béziera przeznaczonymi do modelowania geometrycznego, tylko interpretowane są jako funkcje interpolujące w metodzie klatek kluczowych, czyli jako funkcje opisujące zmianę wartości animowanego parametru w czasie.

5.2 Błąd aproksymacji a wizualny odbiór aproksymowanej trajektorii

W analizach, jakim została poddana prezentowana w pracy metoda, głównym wyznacznikiem jest błąd aproksymacji zdefiniowany w (4.28), który to błąd jest oparty o normę w przestrzeni Sobolewa (opisaną w p. 4.3.3). Wybranie tej normy spowodowało, iż wartość błędu w tej normie jest górnym oszacowaniem wartości błędu aproksymacji w normie supremum (jest to istotne w szczególności z punktu widzenia treści rozdziału 5.2.2).

Dysponując jedynie bezwzględną wartością błędu aproksymacji należy znaleźć zależność między tą wartością a jej wpływem na akceptowalność odbioru aproksymowanej trajektorii przez użytkownika. Poprzez akceptowalną aproksymację rozumie się niezauważalną dla ludzkiej percepcji różnicę pomiędzy trajektorią oryginalną a aproksymowaną. Akceptowalność aproksymacji funkcji, którymi interpolowane są dowolne parametry w metodzie klatek kluczowych,

jest najważniejszym elementem, który powinien być brany pod uwagę w kontekście zagadnień dotyczących animacji komputerowej.

Samo pojęcie akceptowalności aproksymacji trajektorii związane jest szerszym pojęciem percepcji ruchu.

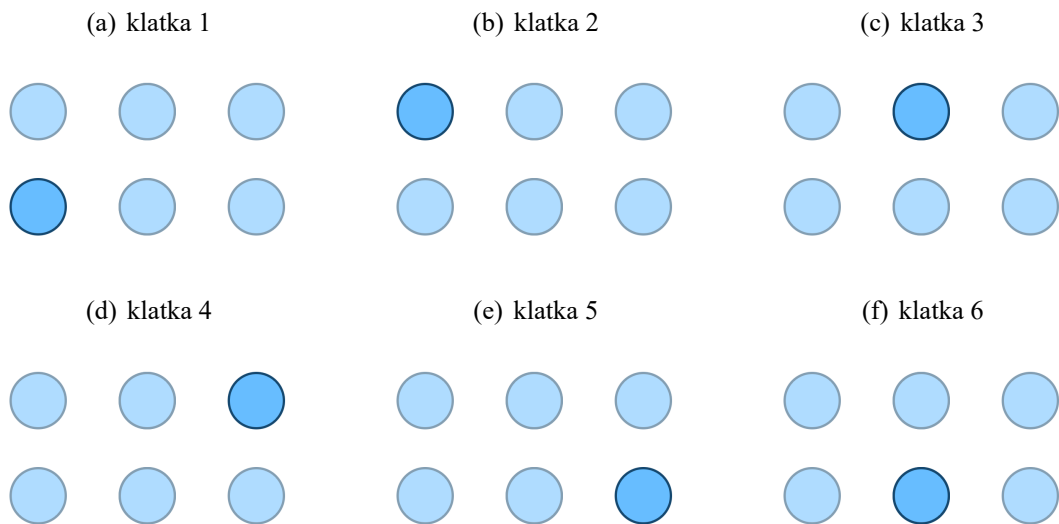
5.2.1 Percepcja ruchu

Percepcja ruchu jest zjawiskiem względności kierunku i prędkości ruchu oraz jego ciągłości, która jest badana przez wiele dyscyplin nauki, takich jak psychologia, neurologia, neurofizjologia, inżynieria oraz informatyka. Chociaż proces ten wydaje się oczywisty dla większości obserwatorów, okazał się on trudnym problemem z perspektywy obliczeniowej, gdyż nauka nie dysponuje pełną teorią widzenia. Jest on przy tym niezwykle trudny do wyjaśnienia pod względem przetwarzania neuronowego.

Nie jest celem tej pracy streszczenie całości wiedzy w zakresie percepcji ruchu, gdyż jej ogrom, gromadzony przez wiele lat przez naukowców badających tę tematykę, wymagałby poświęcenia mu odrębnego opracowania.

Niemniej jednak zasadne jest przedstawienie dwóch pożytecznych pojęć, które w literaturze przedmiotu (np. w [50]) określane są jako:

1. „Percepcja ruchu pierwszego rzędu”. Zjawisko to występuje, gdy dwa lub więcej bodźców, które są naprzemiennie włączane i wyłączane, mogą generować percepcję ruchu. Na rysunku 5.6 przedstawiono ideę „ruchu Beta”, gdzie obiekt postrzegany jest jako poruszający się, gdy w rzeczywistości prezentowana jest seria nieruchomych obrazów. To zjawisko nazywane jest także jako „ruch pozorny” i jest podstawą animacji (w tym filmów). Częstotliwość występowania bodźców musi być odpowiednia. Zbyt szybka spowodowałaby nierozróżnialność bodźców od siebie, a zbyt wolna przerwałaby iluzję ruchu.
2. Pojęcie „percepcji ruchu drugiego rzędu” jest definiowane jako postrzeganie zmiany kontrastu (względnie tekstury lub samo migotanie), która może być postrzegana jako ruch.



Rysunek 5.6: Klatki animacji, które odtwarzane sekwencyjnie po sobie z odpowiednią częstotliwością generują iluzję ruchu.

W tym miejscu wspomnieć trzeba także o eksperymentach Gunnara Johanssona [66][67], których zapis wideo w bardzo dobitny sposób przedstawia jak bardzo złożonym problemem, związanym z przedmiotem niniejszej pracy, jest percepcja ruchu.

Dla problemu aproksymacji trajektorii istotna jest możliwość znalezienia odpowiedzi na pytanie:

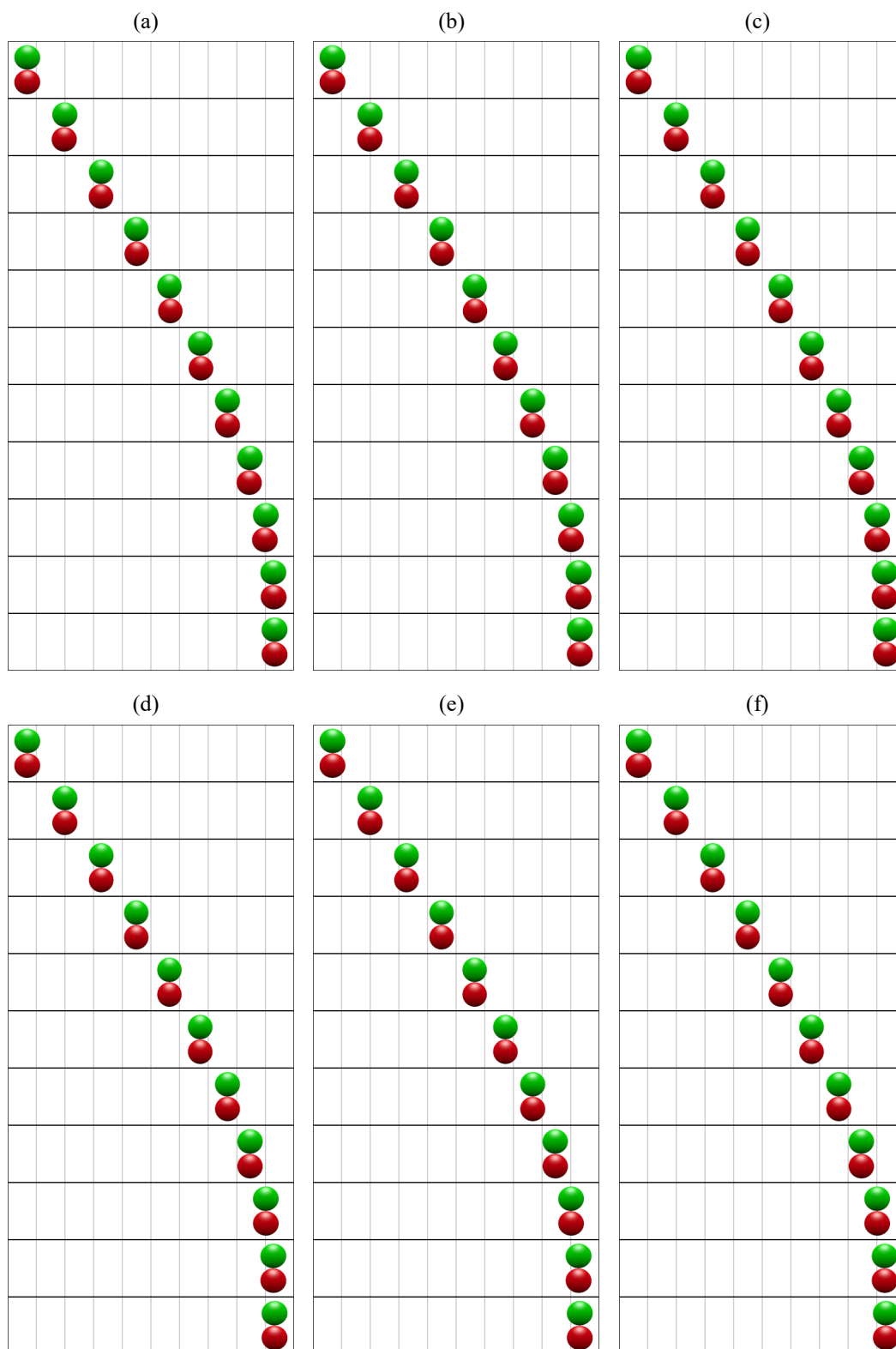
Czy istnieje graniczna wartość błędu aproksymacji (4.28), dla której aproksymowana trajektoria wizualnie nie jest odróżnialna dla odbiorcy od trajektorii wejściowej?

Jeżeli odpowiedź na powyższe pytanie jest twierdząca, to nasuwają się kolejne pytania:

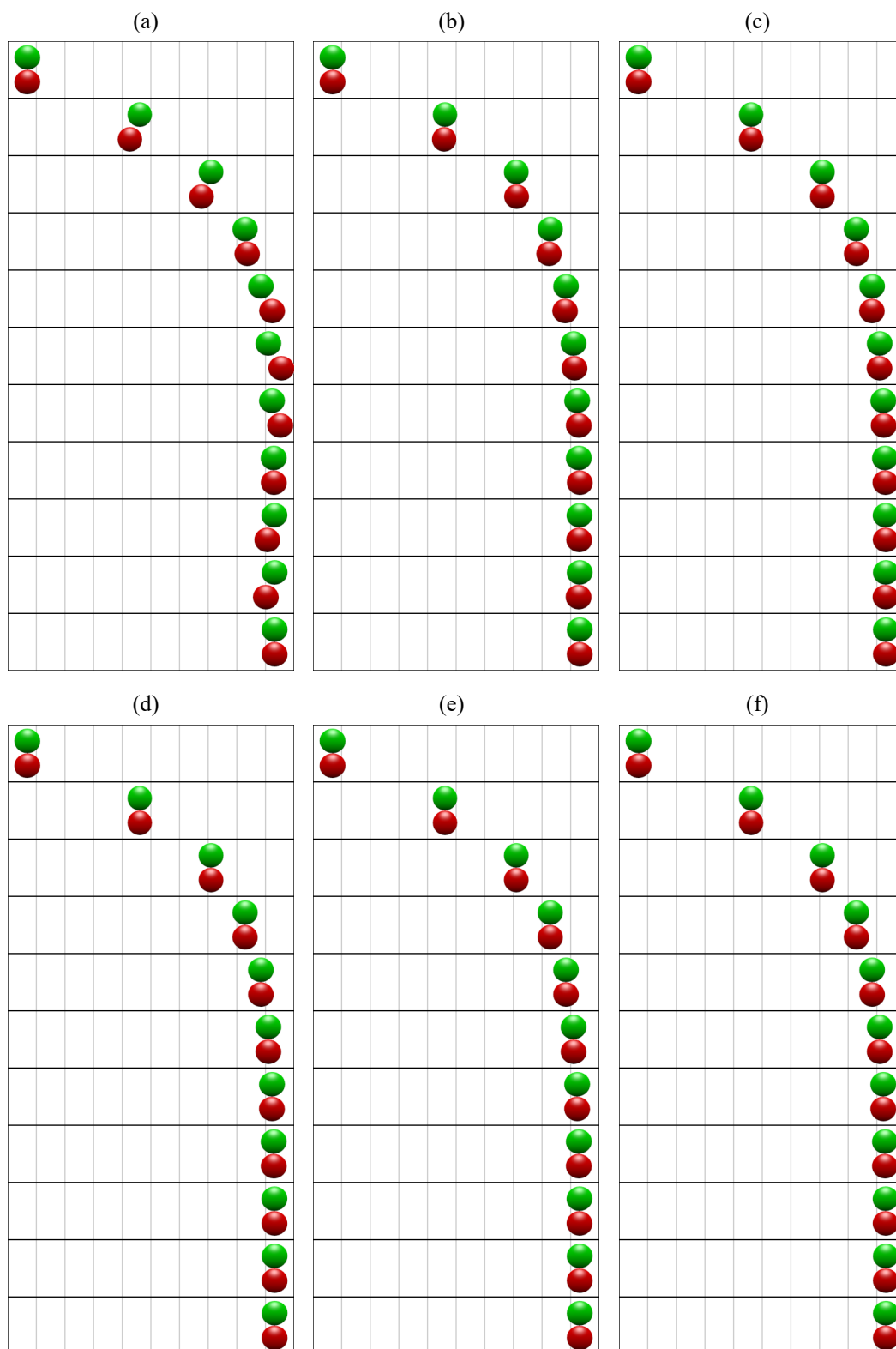
- *Jeżeli istnieje graniczna wartość błędu aproksymacji, to jaką przyjmuje on wartość?*
- *Czy ta wartość jest wspólna dla wszystkich rodzajów parametrów animowanych w metodzie klatek kluczowych?*

5.2.2 Próba wyznaczenia wartości granicznej błędu aproksymacji

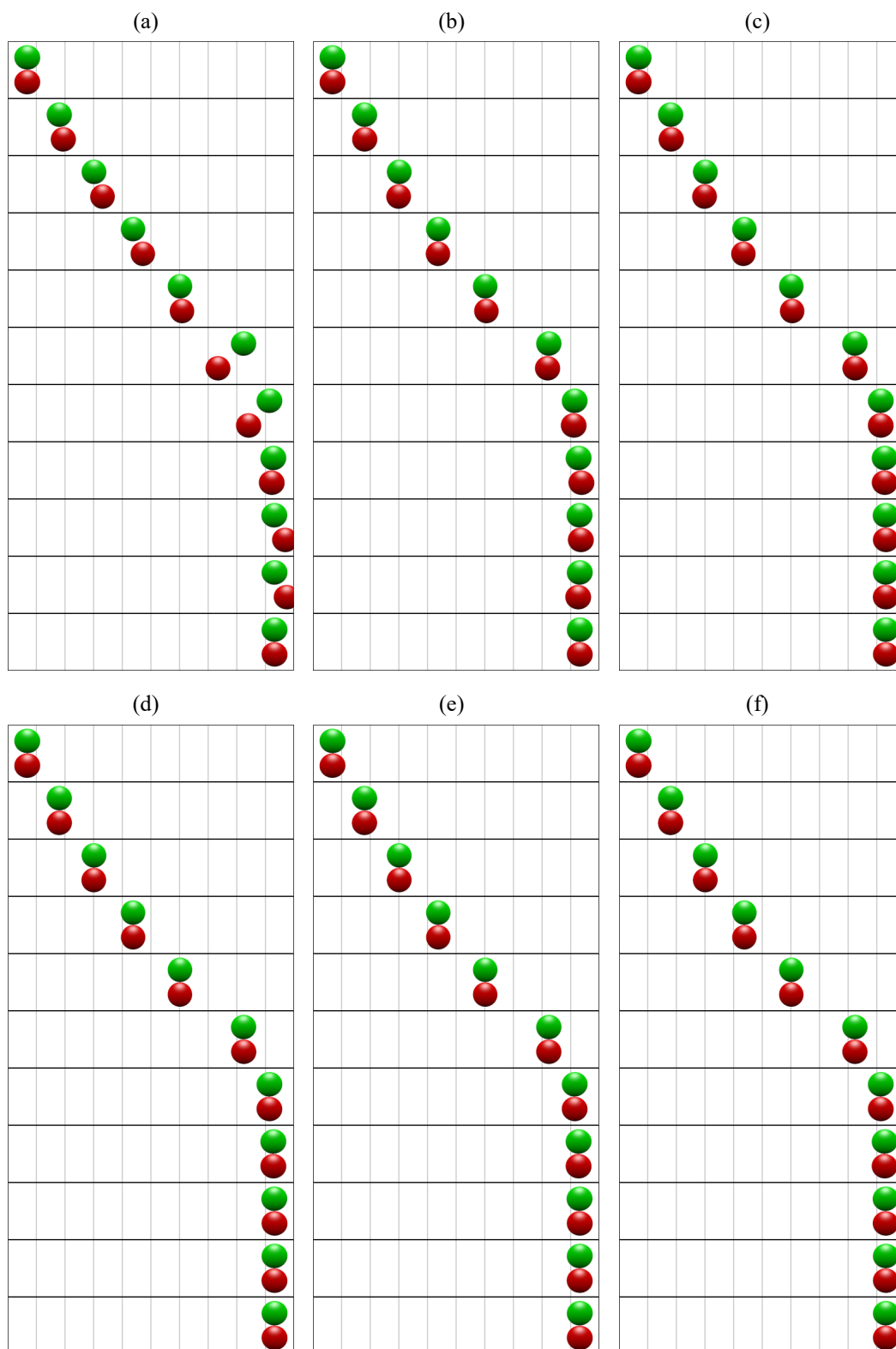
Brak matematycznego modelu ludzkiego widzenia skutkuje tym, że próba odpowiedzi na zasadnicze pytanie o istnienie granicznej wartości błędu aproksymacji wcześniej czy później musi prowadzić do podejścia eksperymentalnego. Na potrzeby badań zostały przygotowane zestawy eksperymentów, w którym testowe krzywe opisane w p. 5.1.3 zostały aproksymowane coraz większą liczbą wielomianów (tj. od jednego aż do sześciu wielomianów co najwyżej trzeciego stopnia). Dla każdego eksperymentu wyznaczono także błąd aproksymacji (4.28). Wyniki eksperymentów przedstawiono na rysunkach od 5.7 do 5.16.



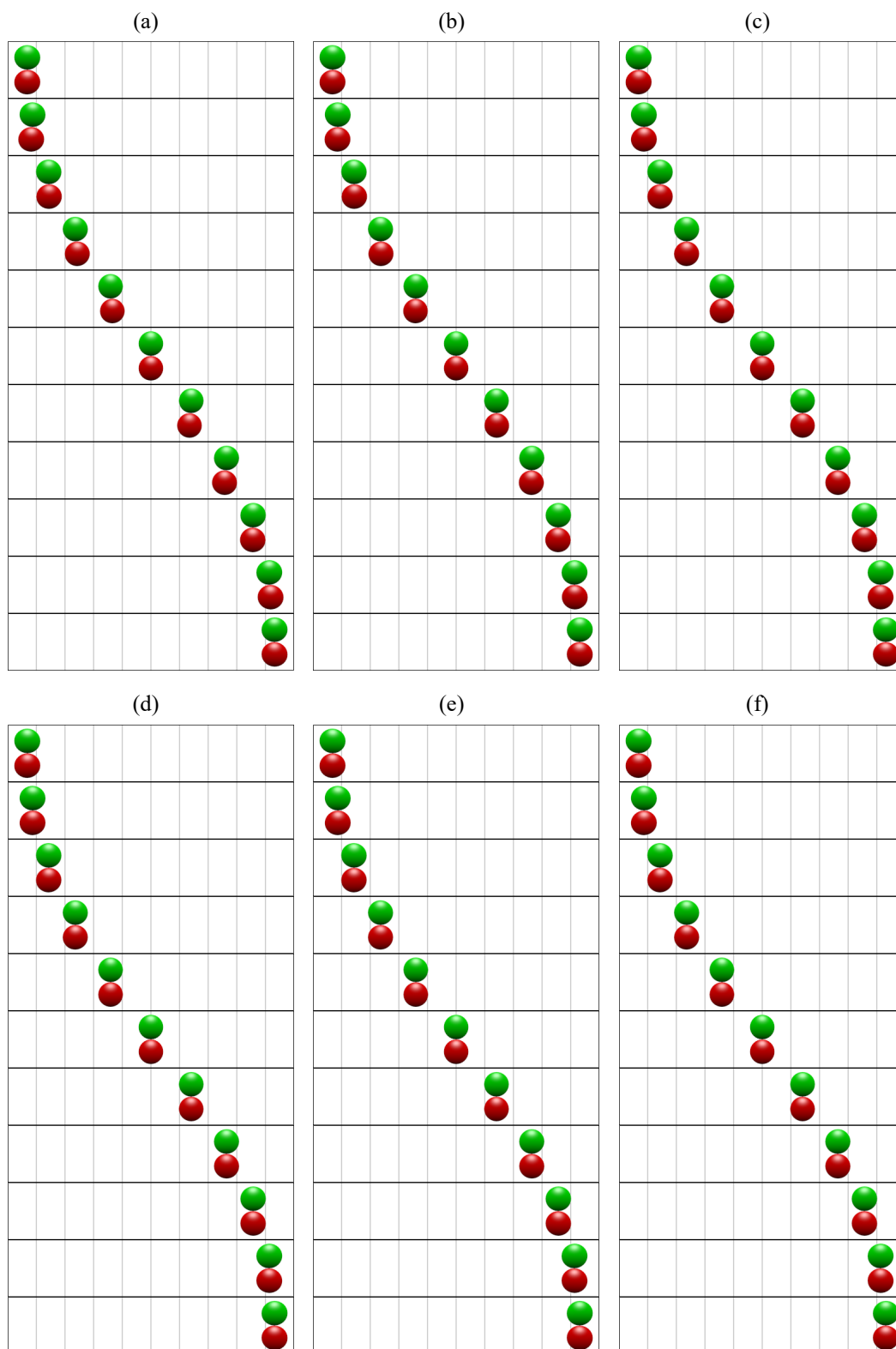
Rysunek 5.7: Klatki animacji trajektorii (Funkcja 1) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 0.332205, (b) 0.139628, (c) 0.020630, (d) 0.012149, (e) 0.005250, (f) 0.003255.



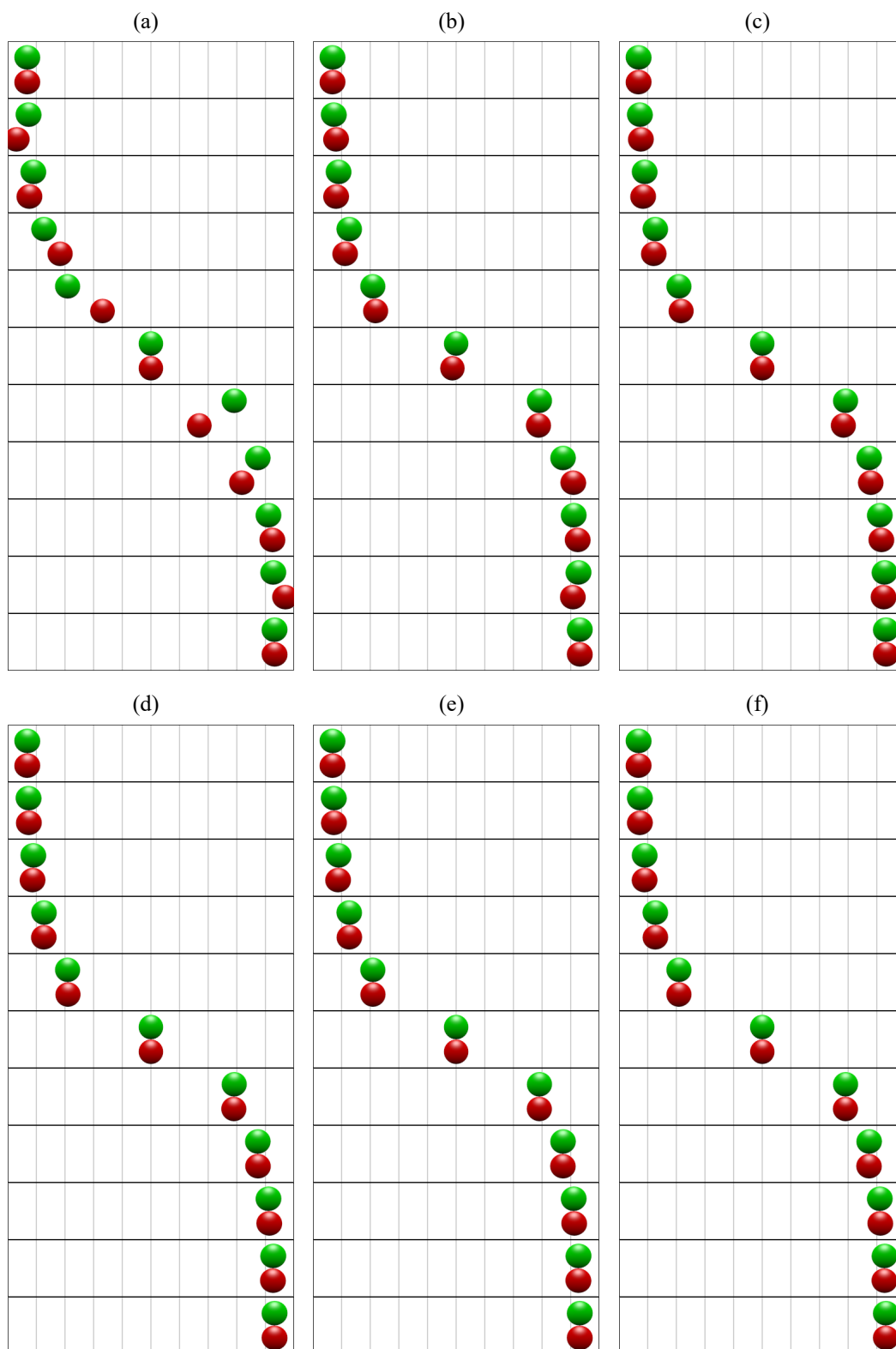
Rysunek 5.8: Klatki animacji trajektorii (Funkcja 2) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 3.155436, (b) 0.762679, (c) 0.174709, (d) 0.047817, (e) 0.030304, (f) 0.017030.



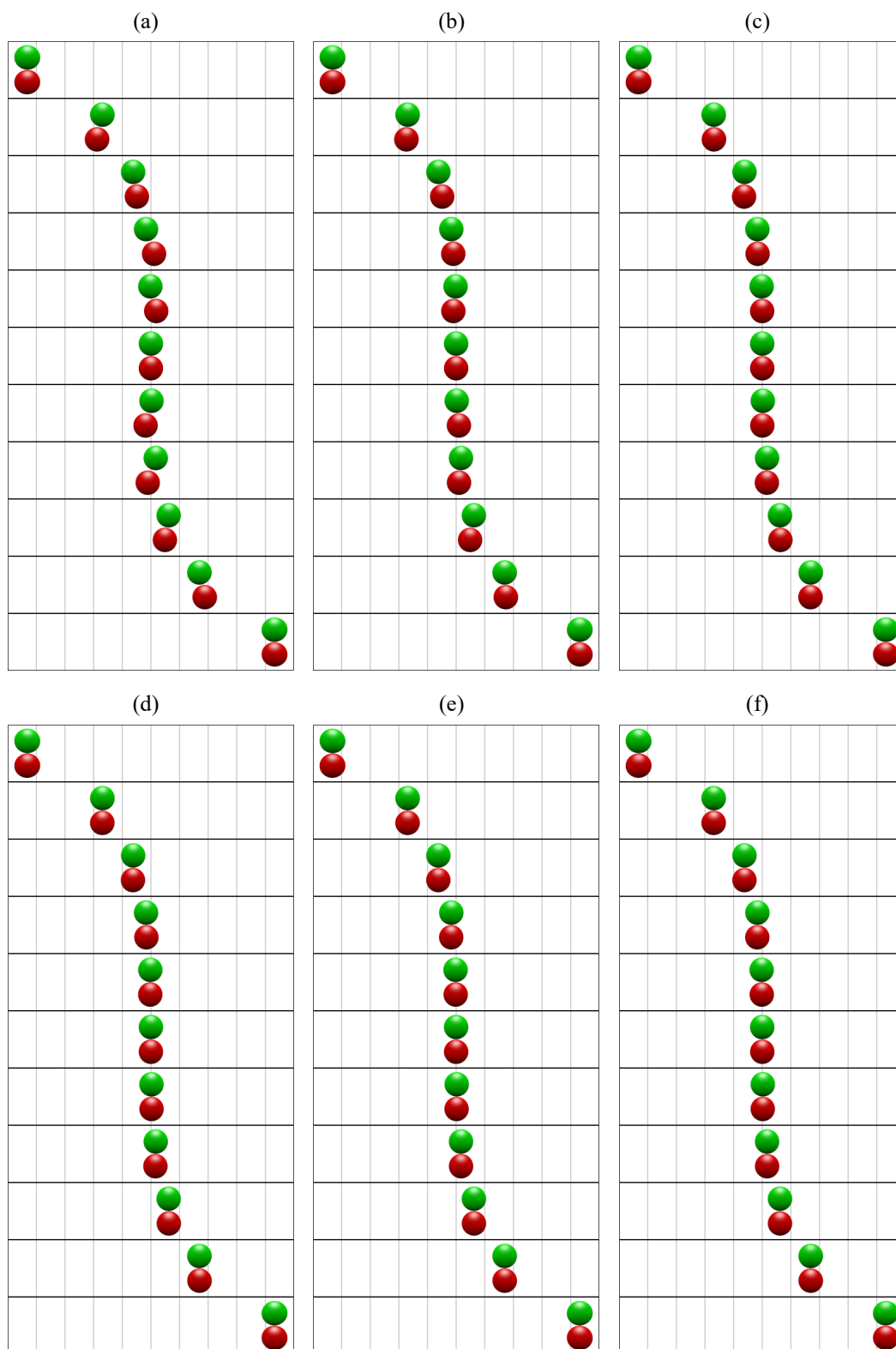
Rysunek 5.9: Klatki animacji trajektorii (Funkcja 3) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 5.550946, (b) 1.480575, (c) 0.447404, (d) 0.227038, (e) 0.126482, (f) 0.055435.



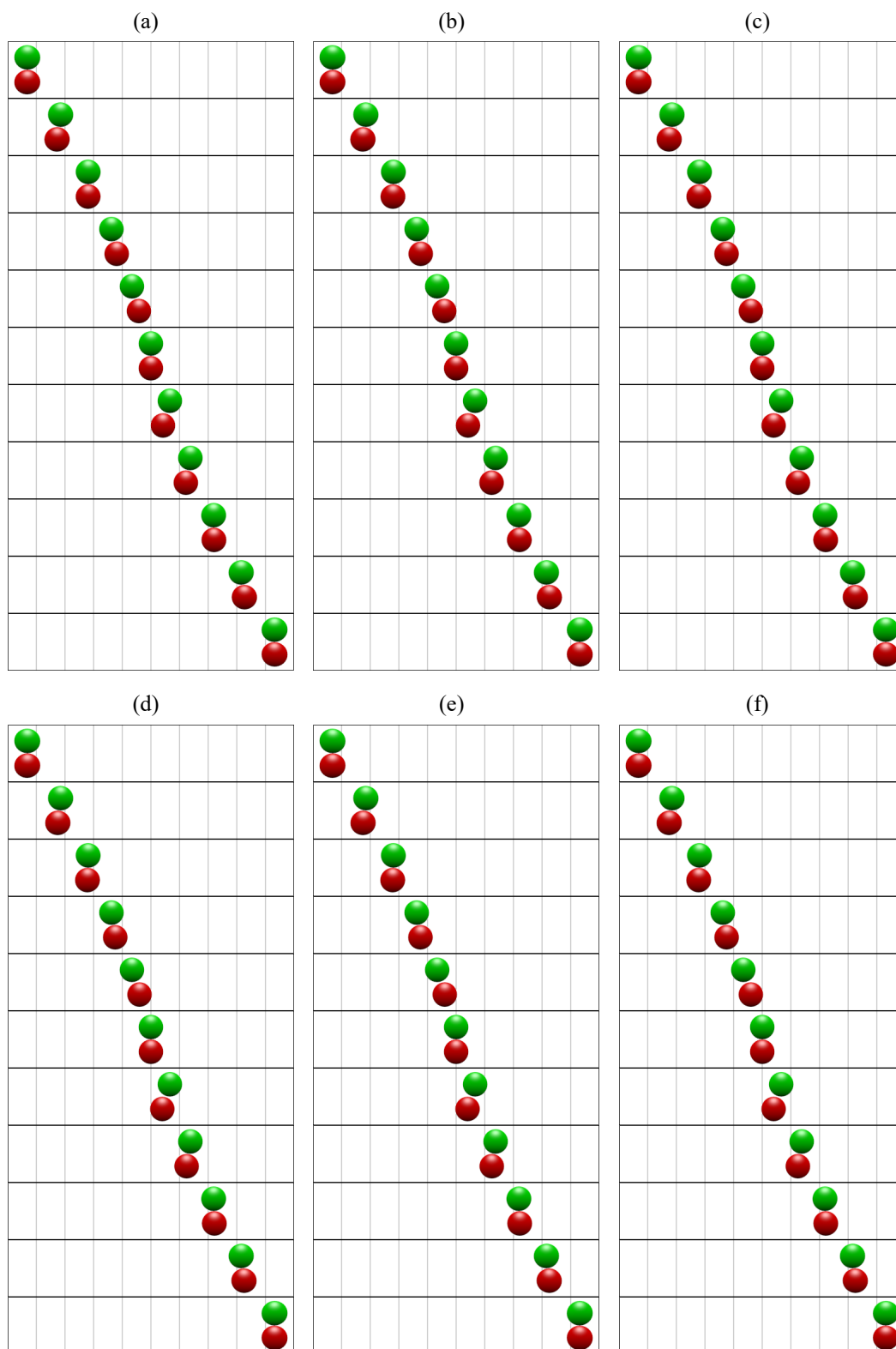
Rysunek 5.10: Klatki animacji trajektorii (Funkcja 4) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowaną w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 0.651050, (b) 0.261670, (c) 0.037146, (d) 0.026938, (e) 0.009444, (f) 0.007517.



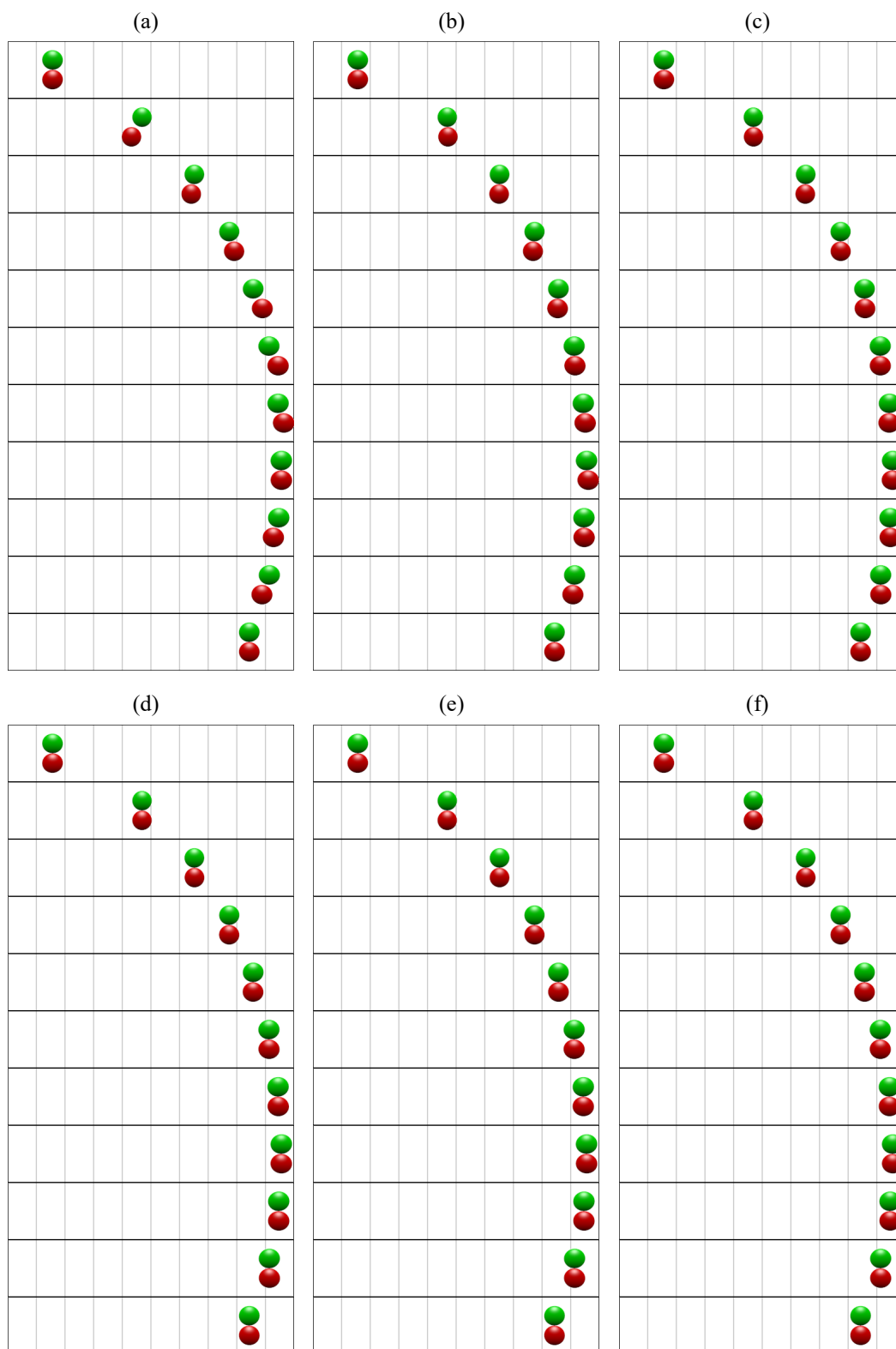
Rysunek 5.11: Klatki animacji trajektorii (Funkcja 5) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 11.228193, (b) 4.709534, (c) 1.856845, (d) 0.609467, (e) 0.448780, (f) 0.325160.



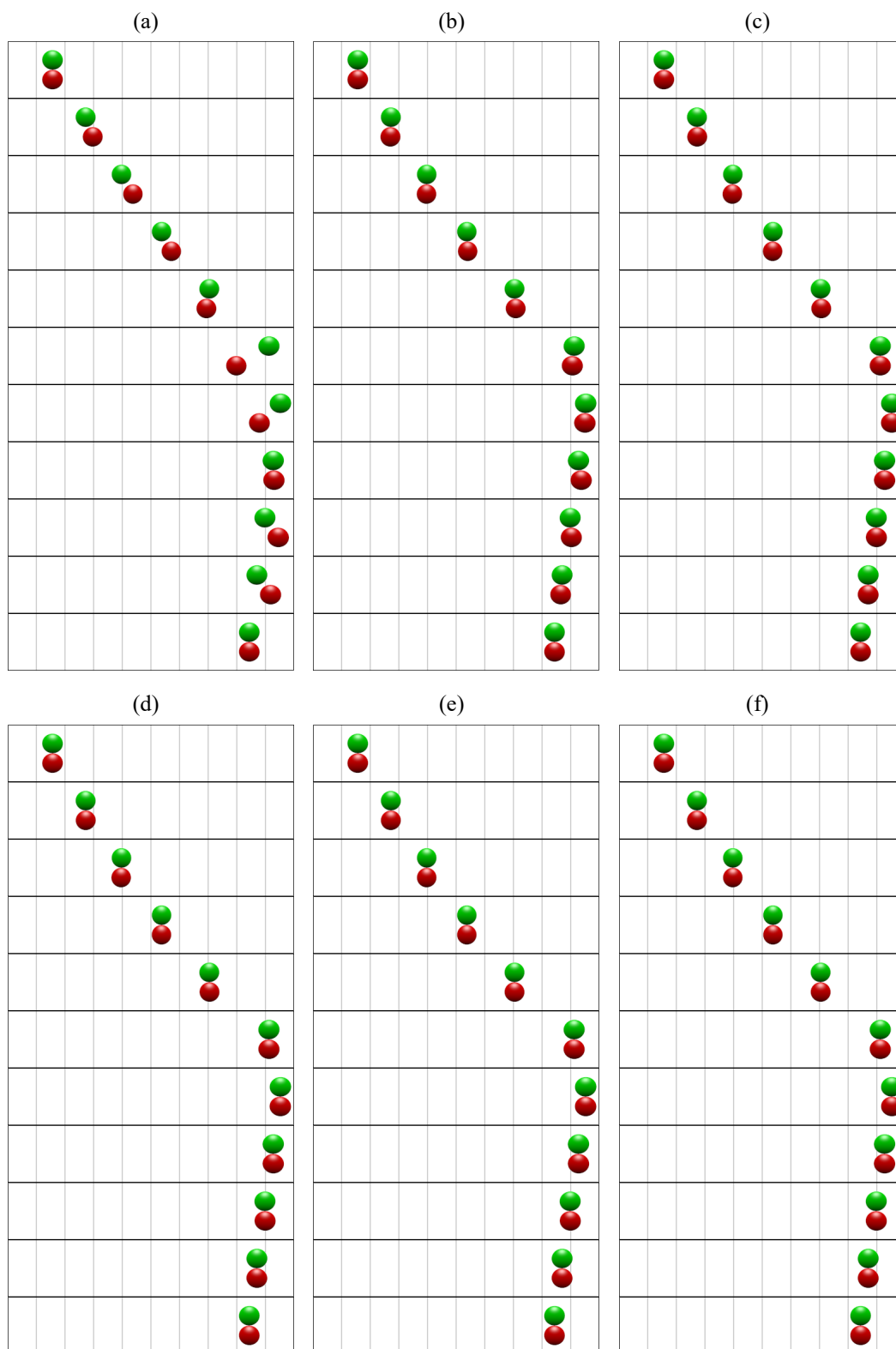
Rysunek 5.12: Klatki animacji trajektorii (Funkcja 6) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 3.265230, (b) 2.231185, (c) 0.434450, (d) 0.320140, (e) 0.116232, (f) 0.090871.



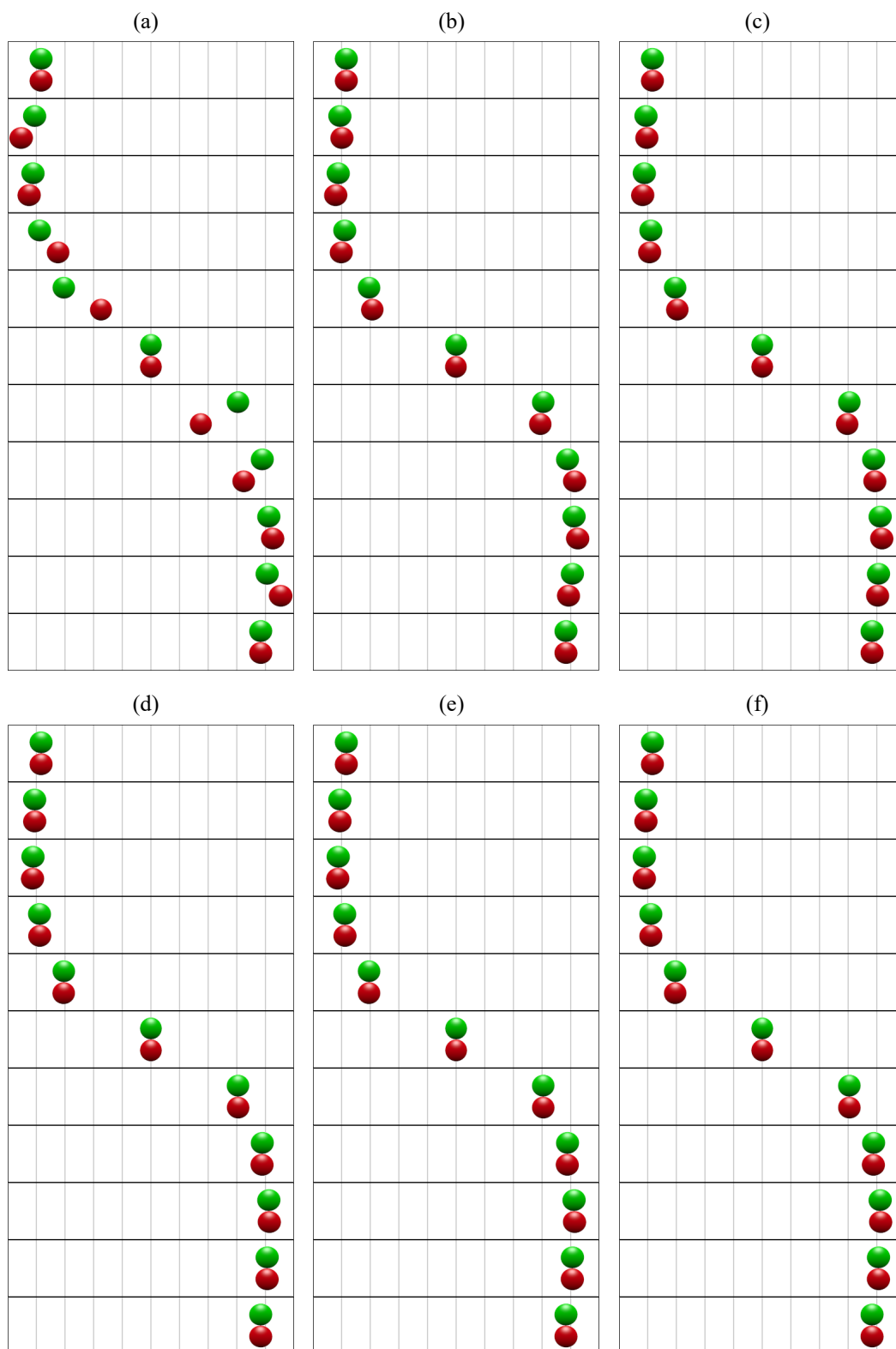
Rysunek 5.13: Klatki animacji trajektorii (Funkcja 7) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 2.245055, (b) 0.737098, (c) 0.304405, (d) 0.102238, (e) 0.069048, (f) 0.050830.



Rysunek 5.14: Klatki animacji trajektorii (Funkcja 8) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 3.775286, (b) 0.918956, (c) 0.385822, (d) 0.166656, (e) 0.090873, (f) 0.055534.



Rysunek 5.15: Klatki animacji trajektorii (Funkcja 9) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 7.0822398, (b) 2.032818, (c) 0.453699, (d) 0.290327, (e) 0.150947, (f) 0.066872.



Rysunek 5.16: Klatki animacji trajektorii (Funkcja 10) wejściowej reprezentowanej przez zieloną kulę, trajektoria aproksymowana metodą prezentowana w pracy to ruch czerwonej kuli. Błąd aproksymacji (a) 12.347805, (b) 4.054040, (c) 1.674225, (d) 0.562308, (e) 0.379762, (f) 0.279563.

Jakie wnioski można wysnuć, analizując eksperymenty przedstawione na rysunkach 5.7 -

5.16? Co z tych eksperymentów, na testowych krzywych, można wywnioskować? Z rysunku 5.7 wynika, że istnieją takie trajektorie, które aproksymowane tylko jednym wielomianem co najwyżej trzeciego stopnia są wizualnie „bardzo podobne” do wejściowej trajektorii. Ale przeważnie aproksymacja tylko jednym wielomianem nie jest wystarczająca. Aby bardziej „upodobnić się” do wejściowej trajektorii (ruchu zielonej kuli), trzeba użyć większej liczby wielomianów aproksymujących.

Czy na podstawie powyższych przykładów istnieje możliwość odpowiedzenia na pytanie: *Czy istnieje graniczna wartość błędu aproksymacji (4.28) dla której trajektoria wyznaczona przez wielomiany aproksymujące wizualnie nie jest odróżnialna dla odbiorcy od trajektorii wejściowej?* Dla wszystkich przeanalizowanych przypadków można uznać że taka graniczna wartość istnieje. Ale tak mała liczba przeanalizowanych przypadków jest niewystarczająca do kategorycznego stwierdzenia, że taka granica istnieje w ogólności (przeanalizowane przypadki stanowią jedynie wskazówkę, że taka granica występuje dla konkretnych przypadków). Aby móc obiektywnie odpowiedzieć na tak zadane pytanie, należałoby przeprowadzić odrębne badania:

- Przeprowadzić oddzielne badania na licznej reprezentatywnej próbie ludzi.
- Przeprowadzić badania na liczniejszej próbce trajektorii, różniących się czasem odtwarzania, rodzajem interpolowanego parametru itp.
- Przeprowadzić badania w różnych warunkach, np. animacjach prezentowanych na monitorze komputera, ekranach kinowych czy też urządzeniach mobilnych.
- Przeprowadzić badania nad wpływem relacji rozmiaru obiektu do rozmiaru ekranu na „akceptowalną” wielkość błędu aproksymacji.

Nie jest znana matematyczna skala porównawcza w zakresie niniejszych badań. Jednocześnie przeprowadzanie takich badań nie jest celem prezentowanej pracy. W ten sam sposób (przeprowadzając oddzielne badania) należałoby poszukiwać odpowiedzi na pozostałe pytania zadane powyżej.

Jako że odpowiedź na główne pytanie o istnienie (oraz o wielkość) błędu aproksymacji, implikuje konieczność przeprowadzania oddzielnych rozległych badań, postawić można pomocnicze pytanie:

Czy istnieje graniczna wartość błędu aproksymacji, dla której postrzeganie (a co za tym idzie akceptacja) trajektorii aproksymowanej odseparowana jest od subiektywnej percepcji ludzkiej?

Cyfrowy charakter wyświetlania animacji na urządzeniach, jakimi są monitory, telewizory czy rzutniki, daje szansę na wyznaczenie wartości błędu aproksymacji w normie supremum, która będzie mniejsza niż najmniejsza „wielkość” możliwa do wyświetlenia na tych urządzeniach. Najmniejsza wielkość rozumiana jest jako np. najmniejsza różnica w odwzorowaniu koloru materiału możliwa do wyświetlania na ekranie monitora, czy też najmniejsza odległość w

wirtualnej przestrzeni, która możliwa jest do zaobserwowania na rastrowym monitorze (więcej na ten temat w p. 5.2.2.1).

Dodatkowym aspektem, który nakłada się na cyfrowy charakter wyświetlania animacji, jest skończona rozdzielczość kąta ludzkiego oka (szerzej opisana w p. 5.2.2.2), która została wyznaczona doświadczalnie i jest powszechnie uwzględniana w zagadnieniach związanych z widzeniem.

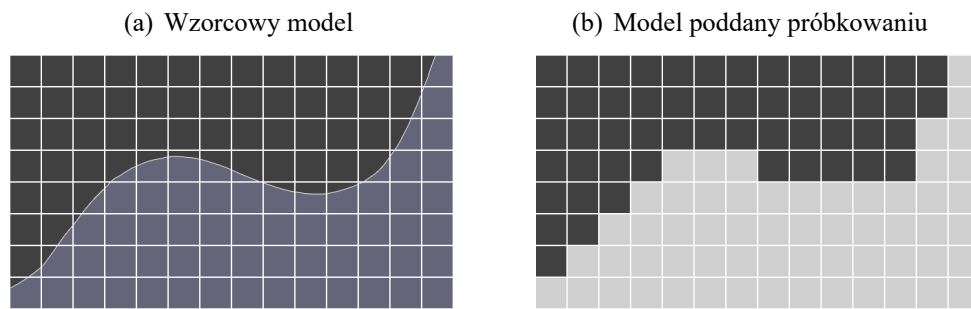
Ponieważ zbieżność ciągu funkcyjnego w normie Sobolewa $W^{1,2}$ na odcinku (a, b) implikuje zbieżność jednostajną, więc błąd aproksymacji mierzony w normie Sobolewa ogranicza z góry błąd aproksymacji w normie supremum z dokładnością do stałej $|a - b|$. Możemy zatem zadając z góry odpowiednią wielkość błędu w normie Sobolewa uzyskać pożądaną wielkość błędu w normie supremum. W celu uzyskania błędu aproksymacji, małego w normie Sobolewa, dokonujemy dostatecznie drobnego podziału odcinka (a, b) , w ten sposób możemy uzyskać wartość błędu w normie supremum mniejszą niż najmniejsza „wielkość” możliwa do wyświetlenia na tych urządzeniach.

Wszystko to powoduje, że problemy percepcyjne w aproksymacji nie są już istotne i różnica między trajektoriami wejściową oraz aproksymowaną nie będzie możliwa do zauważenia podczas odtwarzania na monitorach rastrowych czy też rzutnikach przez obserwatora.

Dalsze badania nad akceptowalnością aproksymowanej trajektorii mogą spowodować powiększenie granicznego błędu wyznaczonego podczas odpowiedzi na pomocnicze pytanie, gdyż nawet niewielkie różnice (w granicach kilku pikseli) między trajektoriami mogą być mało zauważalne albo niewypływające negatywnie na odbiór aproksymowanej trajektorii. Przeprowadzenie oddzielnych badań nad akceptowalnością pozwoliłoby bardziej sprecyzować nową graniczną wartość błędu aproksymacji.

5.2.2.1 Aliasing

Aliasing to nieodwracalne zniekształcenie sygnału w procesie próbkowania wynikające z niespełnienia założeń twierdzenia o próbkowaniu, zwanym także twierdzeniem Kotielnikowa-Shannona [41]. Aliasing stanowi potencjalne źródło błędów, które nie występują przy bezpośrednim przetwarzaniu sygnałów analogowych i jest nierozłącznie związane z przetwarzaniem cyfrowym (np. jak na przedstawionym rysunku 5.17).



Rysunek 5.17: Sygnał analogowy z lewej, oraz poddany próbkowaniu z prawej.

W grafice komputerowej to zjawisko polega na zniekształceniu obrazu w wyniku zbyt małej częstości jego próbkowania w procesie rasteryzacji. Rasteryzacja zachodzi najczęściej podczas wyświetlania obrazu na ekranie, który obecnie najczęściej jest ekranem rastrowym. Może dotyczyć także procesu zamiany modelu obrazu z wektorowego na rastrowy. Rasteryzacja występuje jako jeden z koniecznych etapów potoku graficznego w bibliotekach graficznych wykorzystywanych w grafice trójwymiarowej (takich jak DirectX[®][21], jak i OpenGL[®][35], czy też Vulkan[®][36]).

Przykładem aliasingu w komputerowej grafice rastrowej jest występowanie „schodków” na liniach ukośnych lub obrzeżach brył w obrazach na monitorach rastrowych (w których najmniejszym niepodzielnym elementem jest piksel).

Wartość granicznego błędu aproksymacji ε w normie supremum, dla której błąd będzie mniejszy niż błąd wprowadzony przez aliasing, zależy od rodzaju parametru, który poddawany jest interpolacji w metodzie klatek kluczowych. Przeanalizowane zostaną najpopularniejsze przypadki interpolowanych parametrów:

- Gdy interpolowana jest składowa barw w liniowym modelu przestrzeni RGB, gdzie każda składowa barwy kodowana jest 8-bitową wartością w znormalizowanej przestrzeni $[0, 1]$ interpolowanego parametru. Błąd jest łatwo wyznaczyć i wynosi

$$\varepsilon = \frac{1}{2^8} = 0.00390625 \quad (5.8)$$

a przy 10-bitowej wartości na składową RGB

$$\varepsilon = \frac{1}{2^{10}} = 0.0009765625 \quad (5.9)$$

- Gdy interpolowany jest parametr, którym jest pozycja obiektu w przestrzeni wirtualnej. Wyznaczenie wartości nie jest już takie proste, gdyż wartość ta zależy nie tylko od wielkości ekranu monitora (liczonej w pikselach), ale także od rodzaju użytego rzutowania (równoległego lub perspektywicznego) podczas prezentacji sceny.

- Gdy prezentacja świata odbywa się przy użyciu rzutowania ortogonalnego to wartość graniczna błędu opisuje wzór:

$$\varepsilon = \frac{D_c}{D_m} \quad (5.10)$$

gdzie D_c to rozmiar w pikselach szerokości (wysokości) parametru użytego w rzutowaniu ortogonalnym wirtualnej kamery, D_m to szerokość (wysokość) w pikselach monitora na którym wyświetlana jest animacja.

Przykładowo gdy $D_c = D_m = 1920$ czyli szerokość w pikselach standardu telewizyjnego HDTV (oraz bardzo popularnego standardu rozdzielczości monitorów komputerowych) $\varepsilon = \frac{1920}{1920} = 1$.

Warto także przedstawić błąd graniczny dla znormalizowanej krzywej Béziera, wtedy wartość jego przedstawia wzór

$$\varepsilon_{0,1} = \frac{\varepsilon}{D_c} = \frac{1}{D_m} \quad (5.11)$$

Dla przedstawionego wyżej przykładu wartość błędu wynosi $\varepsilon_{0,1} = \frac{1}{1920} = 0.00052083$.

- Gdy prezentacja świata odbywa się przy użyciu rzutowania perspektywicznego błąd ε zależy dodatkowo od parametrów wirtualnej kamery takich jak: odległość bliskiej płaszczyzny obcinania oraz kąt widzenia kamery. Kolejnym aspektem który wpływa na poszukiwaną wartość jest odległość od kamery obiektu w wirtualnym świecie.

Gdy obiekt porusza się tylko w płaszczyźnie równoległej do płaszczyzny obcinania rzutowania perspektywicznego, błąd graniczny można wyrazić wzorem

$$\varepsilon = \frac{2 \operatorname{tg}\left(\frac{\alpha}{2}\right)z}{D_m} \quad (5.12)$$

gdzie α to kąt widzenia (poziomy lub pionowy) wirtualnej kamery, z to odległość animowanego obiektu do wirtualnej kamery, D_m to szerokość (wysokość) w pikselach monitora na którym wyświetlana jest animacja.

Tę wielkość należy porównywać z największym możliwym przemieszczeniem równoległym do płaszczyzny obcinania oraz oddalonym o z od kamery, które to przemieszczenie w całości widoczne jest w piramidzie widzenia wirtualnej kamery (o kącie widzenia α):

$$x_{max} = 2 \operatorname{tg}\left(\frac{\alpha}{2}\right)z \quad (5.13)$$

I tak dla $\alpha = 90^\circ$ oraz odległości obiektu od kamery $z = 100$ i szerokość ekranu FHD $D_m = 1920$, błąd graniczny wynosi $\varepsilon = 0.104167$, gdzie maksymalne widoczne przemieszczenie to $x_{max} = 200$.

Gdy obiekt nie porusza się jedynie w płaszczyźnie równoległej do bliskiej płaszczyzny obcinania, to wartość graniczna będzie obliczana od płaszczyzny równoległej do bliskiej płaszczyzny obcinania, która jest najbliższa podczas wykonywania tego ruchu obiektu.

Warto także przedstawić błąd graniczny dla znormalizowanej krzywej Béziera, wtedy wartość jego przedstawia wzór

$$\varepsilon_{0,1} = \frac{\varepsilon}{x_{max}} = \frac{1}{D_m} \quad (5.14)$$

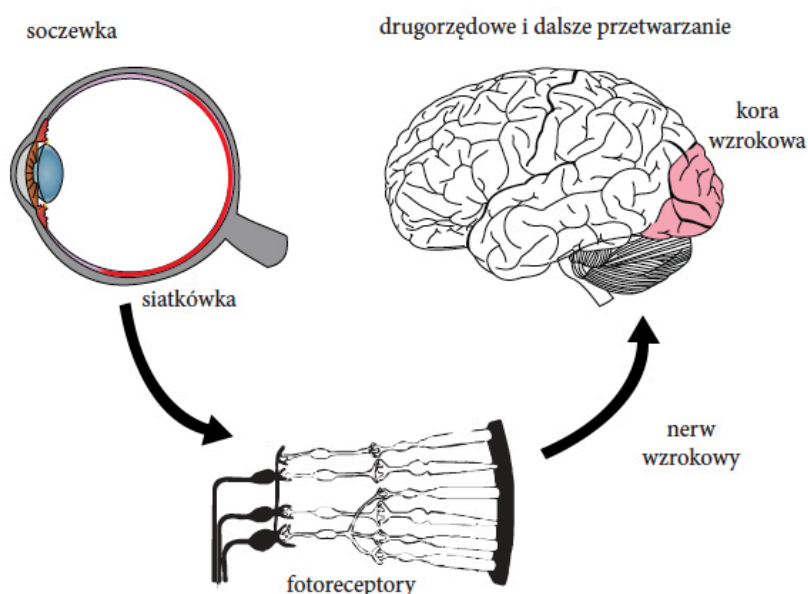
- Gdy interpolowana jest rotacja używająca układu trzech kątów (kątów Eulera) względem osi układu współrzędnych, należy uwzględnić dodatkowo dokładność, z jaką obliczana jest wartość funkcji trygonometrycznych \sin oraz \cos . W zależności od platformy sprzętowej, na której odtwarzana jest animacja, z różną dokładnością obliczane są wartości funkcji trygonometrycznych przez funkcje dostarczane przez biblioteki matematyczne. Dla komputerów osobistych oraz procesorów z rodziny i686 (najczęściej występujących w komputerach osobistych oraz serwerach), ta dokładność wynosi 1 ULP (ang. *unit of least position*) (na podstawie dostępnych danych z [68]), przy założeniu reprezentacji zmiennoprzecinkowej [49]. Przekłada się to na dokładność rzędu $2^{-52} \approx 10^{-15}$ dla liczb zmiennoprzecinkowych podwójnej precyzji. Warto zaznaczyć, że deklarowane dokładności, udostępnione przez producentów układów, nie zawsze mają potwierdzenie w rzeczywistych badaniach (np. przedstawionych w [69]).

Wzór na błąd graniczny jest identyczny ze wzorem dla przypadku przemieszczenia (5.10) lub (5.12) w zależności od użytego rzutowania, powiększony tylko o wartość r , gdzie r to promień sfery opisującej obiekt względem środka jego obrotu. Ten fakt powoduje, że dowolnie mały błąd aproksymacji drogi kątowej (drogi przebytej przez punkt materialny wzdłuż łuku okręgu) skaluje się liniowo względem promienia sfery otaczającej obiekt. Zatem dla największego rozmiaru obiektu wynoszącego połowę wysokości ekranu monitora (540 pikseli) oraz tej samej kamery rzutowania równoległego, jak w przypadku przesunięcia, błąd będzie 540 razy większy.

- Gdy interpolowane jest skalowanie obiektu względem osi współrzędnych, wzory na błąd graniczny tożsame są ze wzorem na błąd wyznaczony dla przesunięć (translacji) obiektów.
- Gdy interpolowane są inne parametry, takie jak np. kąt widzenia kamery, błąd graniczny jest trudno wyznaczalny, gdyż, jak to pokazane zostało we wcześniejszych punktach, kąt kamery wpływa na błędy przesunięcia oraz rotacji.
- Gdy interpolowane są właściwości materiału, np. parametry modelu oświetlenia Blinna–Phonga [3], wówczas wzór na błąd graniczny również jest trudno wyznaczalny.

5.2.2.2 Anatomia narządu wzroku

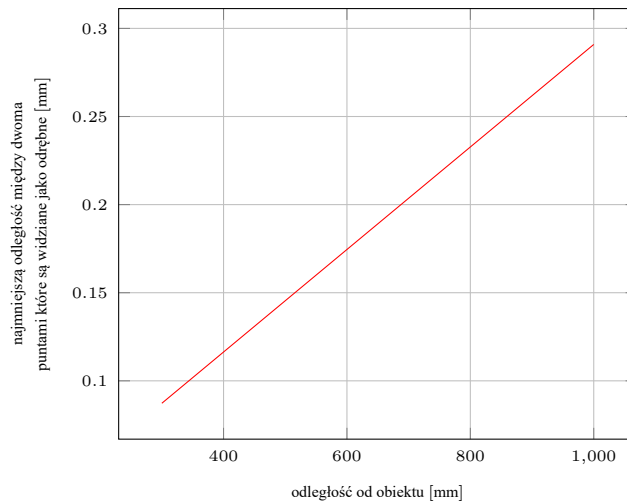
Na negatywne zjawiska występujące przy aliasingu w grafice komputerowej nakłada się także problem ograniczonej rozdzielczości ludzkiego oka. Promienie świetlne dostają się do oka przez źrenicę i soczewkę, a następnie trafiają na siatkówkę na której znajdują się fotoreceptory: pręciki oraz czopki. Te drugie występują w trzech typach o różnej czułości na różne długości fal światła oraz umożliwiają postrzeganie kolorów. Czopki wymagają dobrego oświetlenia do optymalnej pracy, pręciki zaś są bardzo czułe i mogą rejestrować obraz świata przy bardzo słabym oświetleniu. Pozwalają wyłącznie na widzenie monochromatyczne. Użycie dwóch rodzajów receptorów pozwala na rejestrowanie obrazu w zakresie jasności obejmującym dziewięć rzędów wielkości.



Rysunek 5.18: Rysunek przedstawiający poszczególne etapy w procesie widzenia przez człowieka.

Aby dwa punkty mogły być zaobserwowane przez oko jako odrębne obiekty muszą pobudzić dwa różne czopki na siatkówce, oddzielone jednym czopkiem nie pobudzonym. Nazywane jest to zdolnością rozdzielczą siatkówki, która wynosi jedną minutę kątową $1'$. Z zadanej odległości x , od obiektu, można wyznaczyć wzór, który określi najmniejszą odległość między dwoma punktami, które są widziane jako odrębne (zob. rysunek 5.19).

$$d_{min} = 2 \operatorname{tg} \left(\frac{1}{120} \right) x \approx 0.00029088821x \quad (5.15)$$



Rysunek 5.19: Wykres najmniejszej odległości rozróżnianej przez człowieka w zależności od jego odległości od obserwowanego obiektu.

W problemach opisywanych w pracy punkty te można skojarzyć z pikselami w rastrowych monitorach. W takiej sytuacji wzór (5.15) wyznacza minimalną odległość człowieka od monitora z której obserwator nie będzie w stanie dostrzec pojedynczego piksela.

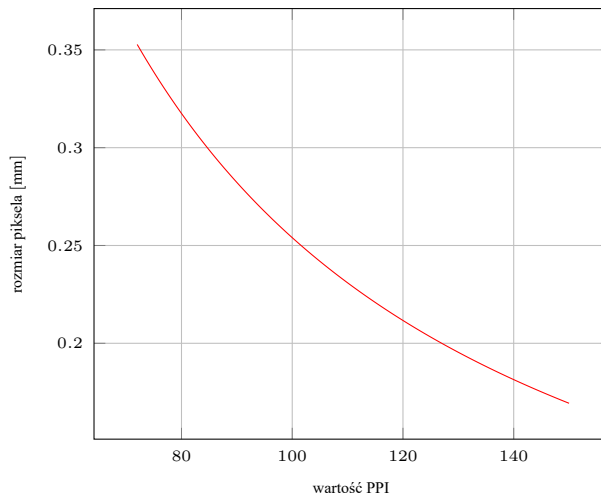
Nie wnikając przy tym głęboko w rzeczywistą budowę ekranu, czy są to monitory ciekłokrystaliczne LCD o różnym typie budowy matrycy, monitory kineskopowe czy też projektory, można założyć, że piksel ma kształt kwadratu. Fizyczną długość boku kwadratu w monitorach (ang. *dot pitch*) można wyznaczyć przy użyciu informacji podawanej przez producenta ekranu, jakim jest parametr PPI (ang. *pixels per inch*), lub też wyznaczyć tę wielkość na podstawie fizycznych rozmiarów ekranu i rozdzielczości ekranu, przy użyciu wzoru

$$ppi = \frac{\sqrt{w_p^2 + h_p^2}}{d_i} \quad (5.16)$$

gdzie w_p wyraża wartość rozdzielczości poziomej monitora wyrażoną w pikselach, h_p to rozdzielczość pionowa, a d_i to przekątna monitora wyrażona w calach.

Długość boku kwadratu reprezentującego piksel wyrażona w milimetrach określa wzór (5.17) - rysunek 5.20

$$d = \frac{25.4}{ppi} \quad (5.17)$$



Rysunek 5.20: Wykres zależności parametru PPI do fizycznego rozmiaru piksela.

5.2.3 Wnioski

Analizując rozważane przypadki w p. 5.2.2.1 oraz w p. 5.2.2.2 można stwierdzić, że nie ma możliwości wyznaczenia jednej granicznej wartości błędu aproksymacji ε dla wszystkich możliwych parametrów użytych w metodzie klatek kluczowych. W zależności od rodzaju użytego parametru wartość graniczna jest inna, albo bezwzględna (np. dla interpolacji kolorów przestrzeni RGB), albo też jest zależna od innych parametrów (jak to ma miejsce w przypadku pozycji obiektu).

Chociaż wartość graniczna dla wszystkich możliwych parametrów użytych w animacji w metodzie klatek kluczowych jest niemożliwa do wyznaczenia, to dla konkretnego przykładu np. dla translacji obiektów przedstawionych na rysunkach od 5.7 do 5.16, wyznaczenie granicznej wartości jest możliwe. Dla parametrów wirtualnej perspektywicznej kamery o wertykalnym kącie widzenia wynoszącym 45° , odległości obiektów od kamery 14.0 oraz szerokości ekranu 1000 pikseli ze wzoru (5.12), błąd graniczny wynosi $\varepsilon = \frac{28 \operatorname{tg}(22.5^\circ)}{1000} \approx 0.011598$. Wartość ta przedstawia błąd w normie supremum i oznacza, że bezwzględna różnica między wartościami trajektorii wejściowej a aproksymowanej poniżej ε nie będzie możliwa do rejestracji przez obserwatora na monitorach komputerowych.

Ta wartość może być większa, gdy uwzględni się dodatkowo odległość, jaką obserwator jest oddalony od monitora komputerowego (wartość ta nie może być pomniejszona, ponieważ obserwator wraz ze zbliżaniem się do monitora będzie coraz bardziej obserwował aliasing występujący w monitorach rastrowych).

Obliczonej wartości błędu $\varepsilon = 0.011598$ nie można bezpośrednio odnieść do wartości błędu przedstawionych w rysunkach od 5.7 do 5.16, gdyż obliczona wartość błędu jest miarą odległości w normie supremum, a w rysunkach przedstawiono wartość błędu w normie $W^{1,2}$. Ale

dzięki właściwościom wybranych norm opisanych w rozdziale 4.3, to w najgorszym przypadku błąd aproksymacji krzywych na rysunkach od 5.7 do 5.16 musi być mniejszy niż ε , by był nie-
możliwy do obserwacji na monitorach rastrowych.

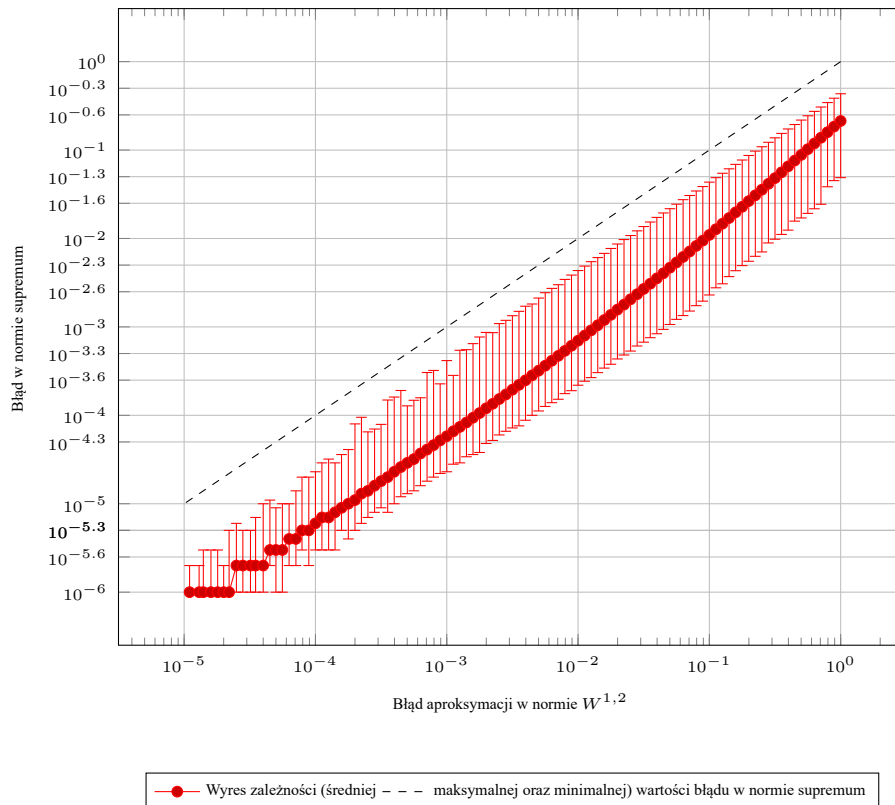
Dokładniejsza analiza oraz testy zależności między tymi błędami aproksymacji w normach supremum a $W^{1,2}$, w zawężonym problemie aproksymacji krzywych Béziere wielomianami co najwyżej trzeciego stopnia, przedstawiono w rozdziale 6.1.

Rozdział 6

Testy oraz analiza rozwiązania

6.1 Błąd aproksymacji w normie $W^{1,2}$ a błąd w normie supremum

Jak zostało pokazane w nierówności (4.24), błąd aproksymacji krzywej Béziera trzeciego stopnia mierzony normą $W^{1,2}$ jest zawsze większy lub równy błędowi w normie supremum z dokładnością stałej równej długości odcinka, na którym dokonywana jest aproksymacja. Jako że długość odcinka postaci znormalizowanej krzywej Béziera podanym testom równa jest 1, warto jest zbadać, jak ta nierówność (4.24) wygląda w zawężonym problemie, ograniczonym do wielomianów co najwyżej trzeciego stopnia oraz ich odwrotności, co jest jednym z przedmiotów badań prezentowej pracy. Do tego badania przygotowano testy. Zbadano aproksymacje krzywej Béziera wykorzystując od 1 do 6 wielomianów w równomiernym podziale względem długości krzywej i wyznaczono dla tej aproksymacji wartości błędów w normie Sobolewa oraz normie supremum. Dla znormalizowanej krzywej Béziera współrzędne punktów kontrolnych, $P1$ oraz $P2$, zawarte zostały w zbiorach $x \in 0.10, 0.15, \dots, 0.85, 0.90$ oraz $y \in -1.0, -0.9, \dots, -0.1, 0.0, 0.05, \dots, 0.95, 1.0, 1.1, \dots, 1.9, 2.0$. Ponadto aproksymowano przy użyciu czterech konfiguracji zachowania ciągłości sąsiadujących krzywych w węzłach wewnętrznych: zachowanie ciągłości C^0 , zachowanie gładkości klasy C^1 , zachowanie gładkości klasy C^2 oraz zachowanie gładkości klasy C^1 wraz zachowaniem gładkości klasy C^1 w węzłach zewnętrznych. Tak zdefiniowane syntetyczne testy wygenerowały ponad 7000000 przypadków znormalizowanych krzywych Béziera, dla których przeprowadzona została analiza porównawcza błędów w dwóch normach (normie supremum oraz normie $W^{1,2}$). Wyniki badań przedstawiono na wykresie 6.1.



Rysunek 6.1: Porównanie błędów normy $W^{1,2}$ względem normy supremum.

Liną przerywaną zaznaczono sytuacje, w której wartością błędu w normie $W^{1,2}$ byłyby równa wartości błędu w normie supremum. Z wyników przedstawionych na wykresie 6.1 można wywnioskować, że wartość błędu aproksymacji w normie supremum względem normy $W^{1,2}$ jest od 2 do 20 razy mniejszy na całej badanej rozpiętości wartości błędu w normie $W^{1,2}$. Średnia wartość błędu w normie supremum w większości przypadków okazała się 10 razy mniejsza niż błąd w normie $W^{1,2}$.

Przekładając te informacje do granicznego błędu testowych krzywych i ich aproksymacji z rysunków od 5.7 do 5.16 z rozdziału 5.2.2, można stwierdzić, że błąd aproksymacji może być dwa razy większy (lub nawet dziesięć razy większy) od wyznaczonej granicznej wartości 0.00603, aby aproksymacja trajektorii była nie do odróżnienia od wejściowej trajektorii przez obserwatora.

6.2 Testy ze względu na liczbę wielomianów aproksymujących funkcję wejściową $C(x)$

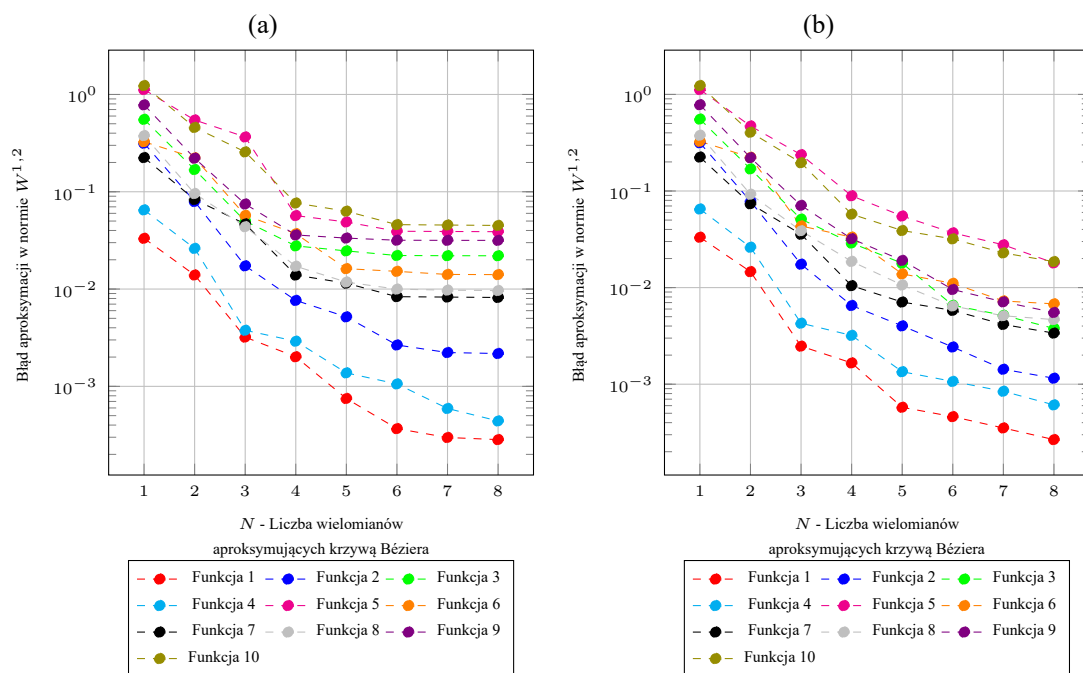
Pierwszym elementem konfiguracji prezentowanej metody jest wybór liczby N wielomianów co najwyżej trzeciego stopnia, którymi aproksymowana jest wejściowa krzywa Béziera trzeciego stopnia. Dodatkowo w proponowanej metodzie siłowej z wyborem liczby N wielomia-

nów bezpośrednio wiąże się problem wyboru wielkości potencjalnych podprzedziałów (stała K), w których testowany będzie błąd aproksymacji funkcji wejściowej. Liczba potencjalnych podprzedziałów (stała M), dla których konieczne jest wyznaczenie błędu aproksymacji wynosi $M = \binom{N}{K}$. Złożoność czasowa takiego podejścia wynosi $O(n) = n!$ co jest bardzo niekorzystne. Z praktycznego punktu widzenia liczba podprzedziałów N oraz stała K musi być ograniczone. Z jednej strony wartość M , w testowych przypadkach, powinna być mniejsza niż 10^9 (by ograniczyć czas potrzebny na wykonanie aproksymacji), z drugiej strony stała N nie powinna przekraczać 7, aby ograniczyć zapotrzebowania pamięciowe dla wielomianów aproksymujących.

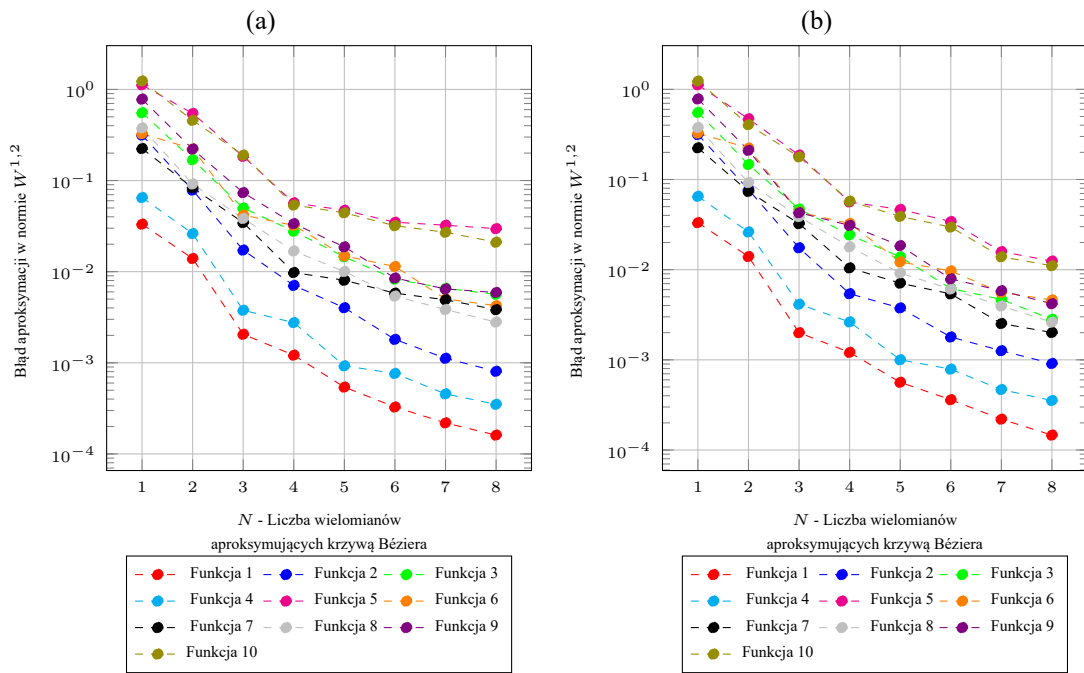
Przetestowane zostały dwa typy podziału równomiernego

1. Typ A - podział równomierny dziedziny funkcji wejściowej $C(x)$.
2. Typ B - podział krzywej wejściowej $C(x)$ na łuki równej długości.

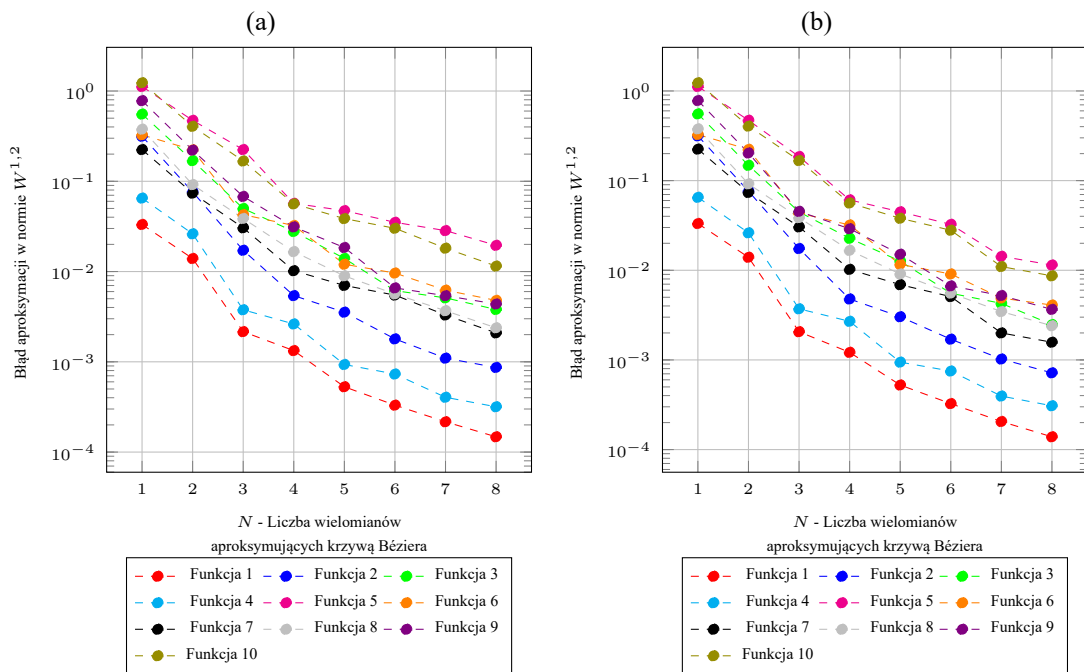
Przedstawienie błędy aproksymacji dla zmiennych N oraz K na jednym wykresie jest kłopotliwe. Dlatego też, ze względu na użyteczność tych wykresów w analizach, zostały przygotowane wykresy w przekrojach dla stałej wartości N i zmiennej K oraz stałej wartości K oraz zmiennej N .



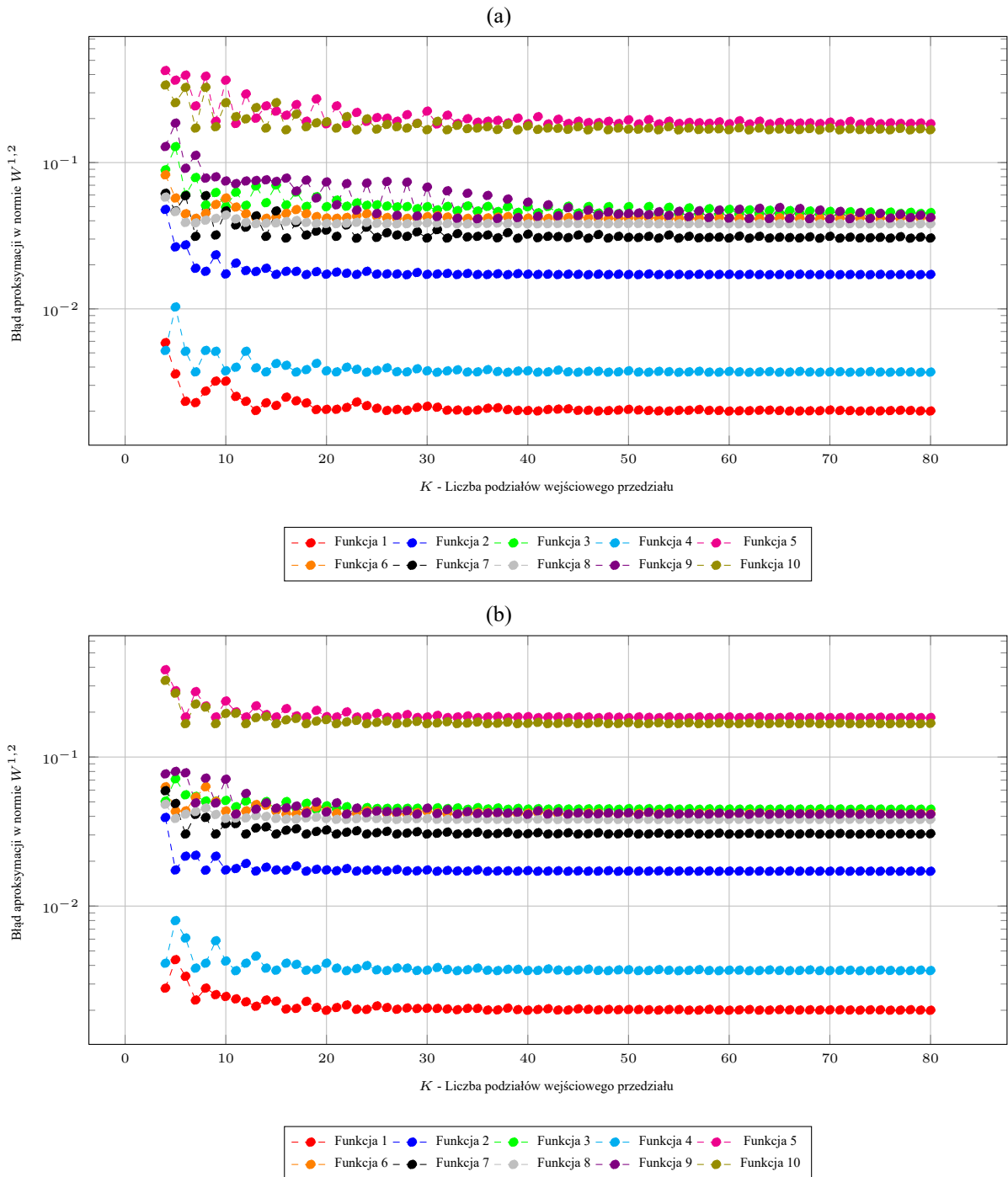
Rysunek 6.2: Wykres błędu aproksymacji testowych krzywych ze względu rodzaj podziału przedziału wejściowego przy $K = 10$. (a) podział równomierny dziedziny funkcji wejściowej $C(x)$, (b) podział krzywej wejściowej $C(x)$ na łuki równej długości.



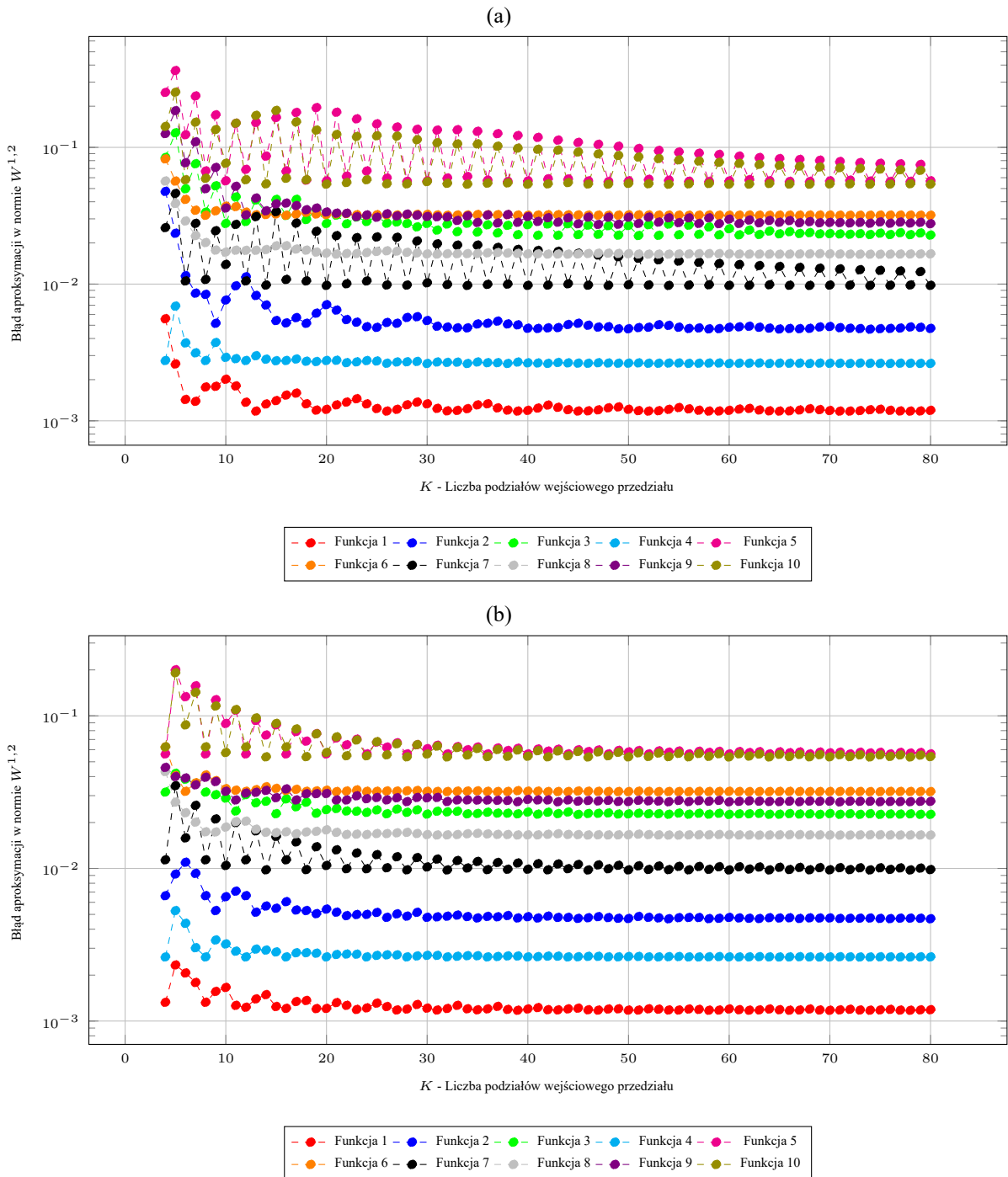
Rysunek 6.3: Wykres błędów aproksymacji testowych krzywych ze względu na rodzaj podziału przedziału wejściowego przy $K = 10$. (a) podział równomierny dziedziny funkcji wejściowej $C(x)$, (b) podział krzywej wejściowej $C(x)$ na łuki równej długości.



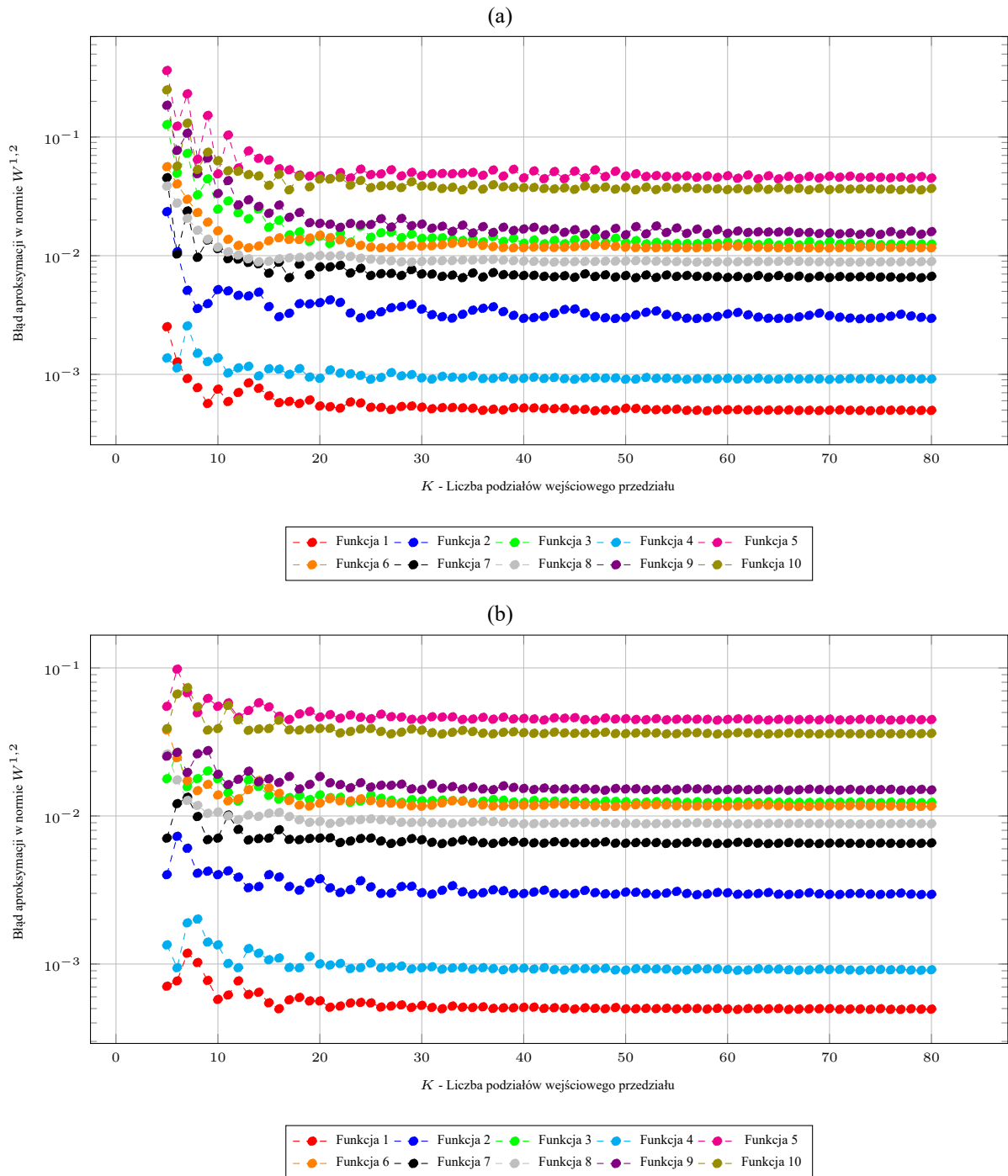
Rysunek 6.4: Wykres błędów aproksymacji testowych krzywych ze względu na rodzaj podziału przedziału wejściowego przy $K = 30$. (a) podział równomierny dziedziny funkcji wejściowej $C(x)$, (b) podział krzywej wejściowej $C(x)$ na łuki równej długości.



Rysunek 6.5: Wykres błędów aproksymacji testowych krzywych ze względu na liczbę potencjalnych podziałów, wykorzystując dwa wielomiany aproksymujące.



Rysunek 6.6: Wykres błędów aproksymacji testowych krzywych ze względu na liczbę potencjalnych podziałów, wykorzystując trzy wielomiany aproksymujące.



Rysunek 6.7: Wykres błędów aproksymacji testowych krzywych ze względu na liczbę potencjalnych podziałów, wykorzystując cztery wielomiany aproksymujące.

Biorąc pod uwagę podział typu A oraz B można stwierdzić, że każdy dodatkowy wielomian użyty w aproksymacji krzywych testowych powodował spadek błędu aproksymacji (4.27), przez co aproksymacja staje się „dokładniejszą”. Ponadto zwiększając liczbę użytych wielomianów, różnica między błędami, dla każdego kolejnego użytego wielomianu, stawała się coraz mniejsza. Pozwala to wnioskować, że błąd aproksymacji w normie $W^{1,2}$ dąży do zera wraz ze wzrostem liczby użytych wielomianów.

Funkcja wejściowa $C(x)$ we wszystkich przypadkach (oprócz jednego) jest funkcją przestępną, dlatego też wielomian co najwyżej trzeciego stopnia pozwala aproksymować wejściową funkcję, w całym przedziale, w większości przypadków ze zbyt małą dokładnością. Użycie większej liczby wielomianów do aproksymacji, wejściowej funkcji (co skutkuje zmniejszeniem przedziału aproksymacji), jest skuteczne, gdyż każdemu wielomianowi, w swoim przedziale, łatwiej jest „dopasować” się do wejściowej funkcji pod względem wartości funkcji oraz wartości pochodnej.

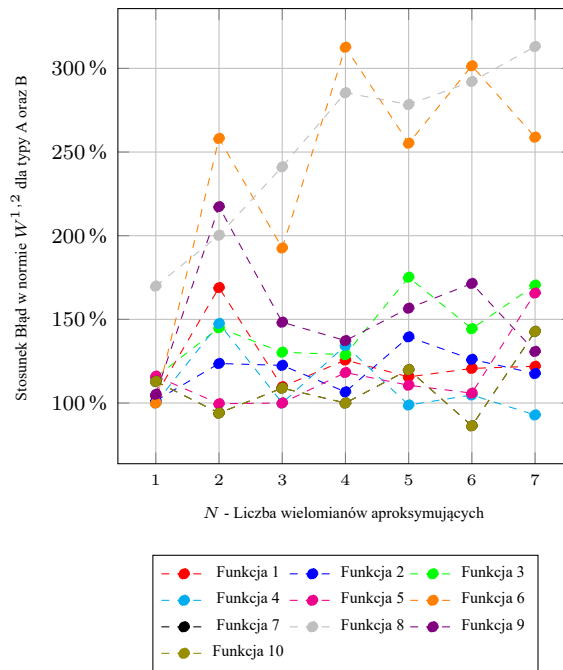
Podział równomierny typu B (podział krzywej wejściowej $C(x)$ na łuki równej długości), względem typu A, powoduje mniejsze oscylacje błędu aproksymacji (jak na rysunku 6.6) przy poszukiwaniu najlepszego rozmieszczenia węzłów aproksymacji, także wraz z wzrostem liczby wielomianów aproksymacyjnych. Typ B podziału powoduje „bardziej liniowy” spadek błędu aproksymacji (przy testowanych wartościach parametru N oraz K). Dla stałej $K < 10$ typ B pozwala wyznaczyć węzły, dla których błąd aproksymacji jest mniejszy od błędu dla typu A. Dla stałej $K > 50$ wybór typu podziału równomiernego A lub B dla większości krzywych testowych przestaje być istotny.

Analizując rysunki 6.5, 6.6 oraz 6.7, można wywnioskować, że znajdowanie najlepszych podprzedziałów jest to problemem nieliniowy, gdyż zwiększenie liczby węzłów aproksymacji w przedziale $[1, 30]$ nie zawsze prowadzi do znalezienia globalnie najmniejszego błędu aproksymacji. Gdy liczba potencjalnych podprzedziałów wynosi powyżej 50, błąd aproksymacji zmniejsza się już w niewielkim stopniu, czyli znalezione węzły leżą już dostatecznie blisko węzłów, dla których błąd osiąga globalne minimum.

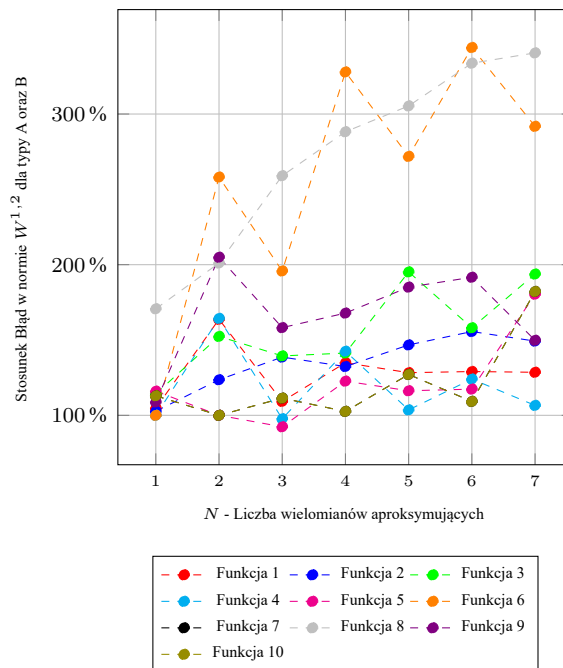
Chociaż z niniejszej analizy nie wynika, że zwiększenie stałej K powyżej 80 pozwala wyznaczyć globalne minimum mniejsze niż prezentowane, jednak poszukiwanie takowego minimum dla większych stałych K zwiększyłoby nadmiernie złożoność obliczeniową.

6.2.1 Porównanie błędów aproksymacji przy podziale równomiernym do podziału nierównomiernego metodą siłową

Warto sprawdzić, jak poszukiwanie najlepszego podziału przy użyciu proponowanej metody siłowej (o złożoności czasowej $O(n) = n!$) wypada na tle równomiernego podziału typu A oraz B. Wyniki takiej analizy przedstawiono na rysunkach 6.8 oraz 6.9.



Rysunek 6.8: Wykres stosunku błędu aproksymacji testowych krzywych przy równomiernym podziale typu B i przy metodzie siłowej dla tego samego typu przy stałej $K = 20$.



Rysunek 6.9: Wykres stosunku błędu aproksymacji testowych krzywych przy równomiernym podziale typu B i przy metodzie siłowej dla tego samego typu przy stałej $K = 30$.

Gdy globalne minimum błędu aproksymacji wyznaczają węzły, które znajdują się w okolicach punktów podziału równomiernego dla typu B, wówczas metoda siłowa (co za tym idzie nierównomierny podział wejściowego przedziału) zmniejsza błąd aproksymacji maksymalnie o około 50%.

Istnieją jednak krzywe (na przykład krzywe testowe 6 oraz 8), dla których błąd aproksymacji dla jednolitego podziału jest ponad dwukrotnie większy niż błąd aproksymacji w podziale nierównomiernym. Zatem poszukiwanie podziału nierównomiernego (np. metodą siłową) przy zwiększeniu złożoności czasowej do jego wyznaczenia jest opłacalny ze względu na dokładność aproksymacji, przy użyciu jak najmniejszej liczby wielomianów aproksymujących.

6.3 Testy błędów aproksymacji w warunkach ciągłości w węzłach

Warunki ciągłości w węzłach są kolejnymi parametrami, które wpływają na realizację prezentowanej metody. Przeprowadzono analizę, jak poszczególne warunki ciągłości wpływają na błąd aproksymacji względem błędu minimalizacji samej funkcji (4.28) (nazywany później **błędem bazowym**).

Jak wiadomo, K -ta pochodna wielomianu stopnia K jest już funkcją stałą. W przypadku wielomianów stopnia co najwyżej trzeciego możemy rozpatrywać warunki gładkości w klasie co najwyżej C^2 w węzłach wewnętrznych oraz zewnętrznych. Podzielono ten aspekt badań na poniższe przypadki możliwych kombinacji zachowania ciągłości:

1. Minimalizacja funkcji bez zachowania jakichkolwiek ciągłości. Ten przypadek jest punktem odniesienia dla pozostałych testów (błąd bazowy). W tabeli 6.1 ta konfiguracja ma numer I.
2. Warunek konieczny do zachowania ciągłości C^0 w węzłach wewnętrznych oraz wartości w węzłach zewnętrznych równe wartości funkcji wejściowej. W tabeli 6.1 ta konfiguracja ma numery II oraz III.
3. Warunek konieczny do zachowania gładkości klasy C^1 w węzłach wewnętrznych. W tabeli 6.1 ta konfiguracja ma numery od IV oraz V.
4. Warunek konieczny do zachowania gładkości klasy C^2 w węzłach wewnętrznych. W tabeli 6.1 ta konfiguracja ma numery od VIII oraz XI.
5. Rozszerzenie warunku z punktu 2 o zadanie konkretnej wartości pochodnych pierwszego oraz drugiego rzędu w węzłach zewnętrznych. W tabeli 6.1 ta konfiguracja ma numery od XII oraz XV.
6. Rozszerzenie warunku z punktu 3 o zadanie konkretnej wartości pochodnych pierwszego oraz drugiego rzędu w węzłach zewnętrznych. W tabeli 6.1 ta konfiguracja ma numery od XVI oraz XXII.

7. Rozszerzenie warunku z punktu 4 o zadanie konkretnej wartości pochodnych pierwszego oraz drugiego rzędu w węzłach zewnętrznych. W tabeli 6.1 ta konfiguracja ma numery od XXIII oraz XXVII.

Zestawienie powyższych warunków zostało przedstawione w tabeli (6.1), gdzie przyjęto następujące oznaczenie kolumn

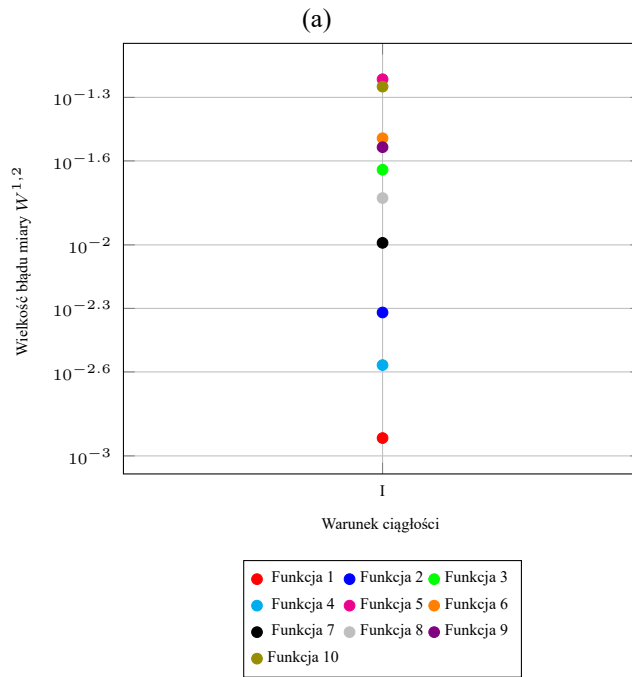
- $C_0 oV$ oznacza że $f_0(s_0) = C(s_0)$ oraz $f_{N-1}(s_N) = C(s_N)$.
- $C_0 cV$ oznacza zachowanie ciągłości C^0 na punktach s_1, \dots, s_{N-1} .
- $C_0 iV$ oznacza $f_i(s_{i+1}) = f_{i+1}(s_{i+1}) = C(s_{i+1})$, $i = 0, \dots, N - 2$.
- $C_1 oV$ oznacza że $f_0(s_0) = C(s_0) \wedge f'_0(s_0) = C'(s_0)$ oraz $f_{N-1}(s_N) = C(s_N) \wedge f'_{N-1}(s_N) = C'(s_N)$.
- $C_1 cV$ oznacza zachowanie gładkości klasy C^1 w węzłach wewnętrznych.
- $C_1 iV$ oznacza $f_i(s_{i+1}) = f_{i+1}(s_{i+1}) = C(s_{i+1}) \wedge f'_i(s_{i+1}) = f'_{i+1}(s_{i+1}) = C'(s_{i+1})$, $i = 0, \dots, N - 2$.
- $C_2 oV$ oznacza że $f_0(s_0) = C(s_0) \wedge f'_0(s_0) = C'(s_0) \wedge f''_0(s_0) = C''(s_0)$ oraz $f_{N-1}(s_N) = C(s_N) \wedge f'_{N-1}(s_N) = C'(s_N) \wedge f''_{N-1}(s_N) = C''(s_N)$.
- $C_2 cV$ oznacza zachowanie gładkości klasy C^2 w węzłach wewnętrznych.
- $C_2 iV$ oznacza $f_i(s_{i+1}) = f_{i+1}(s_{i+1}) = C(s_{i+1}) \wedge f'_i(s_{i+1}) = f'_{i+1}(s_{i+1}) = C'(s_{i+1}) \wedge f''_i(s_{i+1}) = f''_{i+1}(s_{i+1}) = C''(s_{i+1})$, $i = 0, \dots, N - 2$.

Znak \checkmark użyty w kolumnie tabeli (6.1) oznacza, że analizowany przypadek ciągłości (wiersz w tabeli) zawiera konkretny warunek ciągłości.

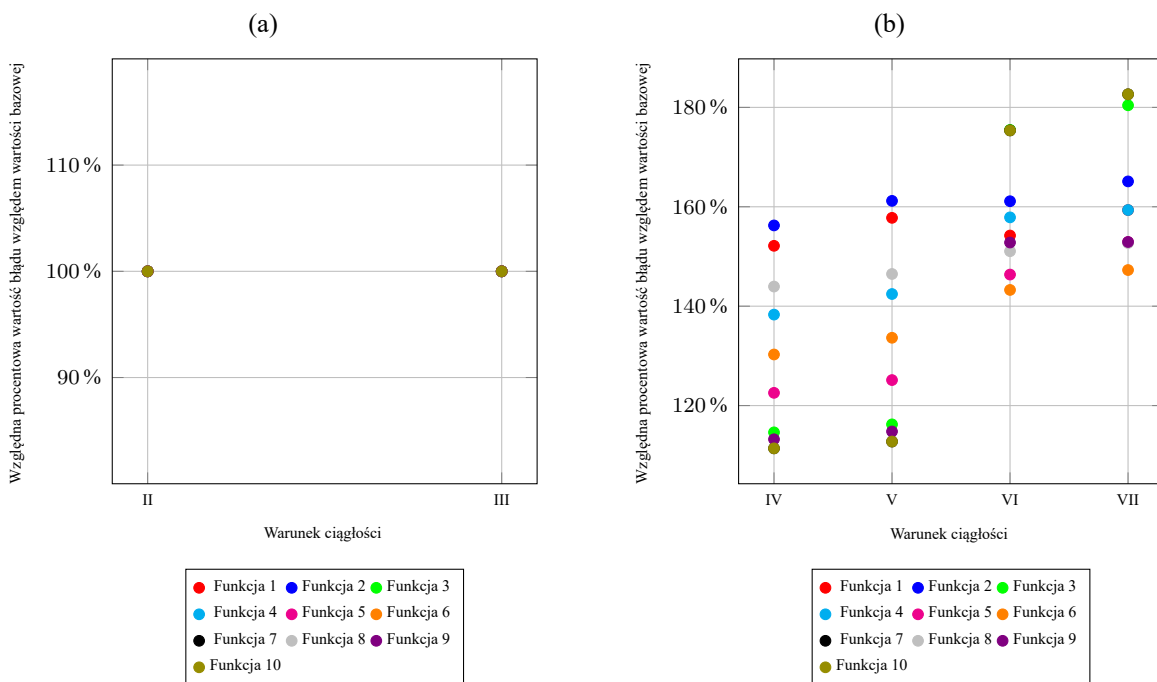
<i>Warunek</i>	<i>Nr.konf</i>	$C_0 oV$	$C_0 cV$	$C_0 iV$	$C_1 oV$	$C_1 cV$	$C_1 iV$	$C_2 oV$	$C_2 cV$	$C_2 iV$
1.	<i>I</i>									
2.	<i>II</i>	✓	✓							
2.	<i>III</i>	✓		✓						
3.	<i>IV</i>	✓	✓			✓				
3.	<i>V</i>	✓		✓		✓				
3.	<i>VI</i>	✓	✓				✓			
3.	<i>VII</i>	✓		✓			✓			
4.	<i>VIII</i>	✓	✓			✓			✓	
4.	<i>IX</i>	✓		✓		✓			✓	
4.	<i>X</i>	✓	✓			✓				✓
4.	<i>XI</i>	✓	✓				✓		✓	
5.	<i>XII</i>	✓	✓		✓					
5.	<i>XIII</i>	✓		✓	✓					
5.	<i>XIV</i>	✓	✓		✓			✓		
5.	<i>XV</i>	✓		✓	✓			✓		
6.	<i>XVI</i>	✓	✓		✓	✓				
6.	<i>XVII</i>	✓		✓	✓	✓				
6.	<i>XVIII</i>	✓	✓		✓		✓			
6.	<i>XIX</i>	✓		✓	✓		✓			
6.	<i>XX</i>	✓	✓		✓	✓		✓		
6.	<i>XXI</i>	✓		✓	✓	✓		✓		
6.	<i>XXII</i>	✓	✓		✓		✓	✓		
7.	<i>XXIII</i>	✓	✓		✓	✓			✓	
7.	<i>XXIV</i>	✓		✓	✓	✓			✓	
7.	<i>XXV</i>	✓	✓		✓	✓				✓
7.	<i>XXVI</i>	✓	✓		✓		✓		✓	
7.	<i>XXVII</i>	✓	✓		✓	✓		✓	✓	

Tablica 6.1: Zestawienie badanych kombinacji zachowania ciągłości w węzłach.

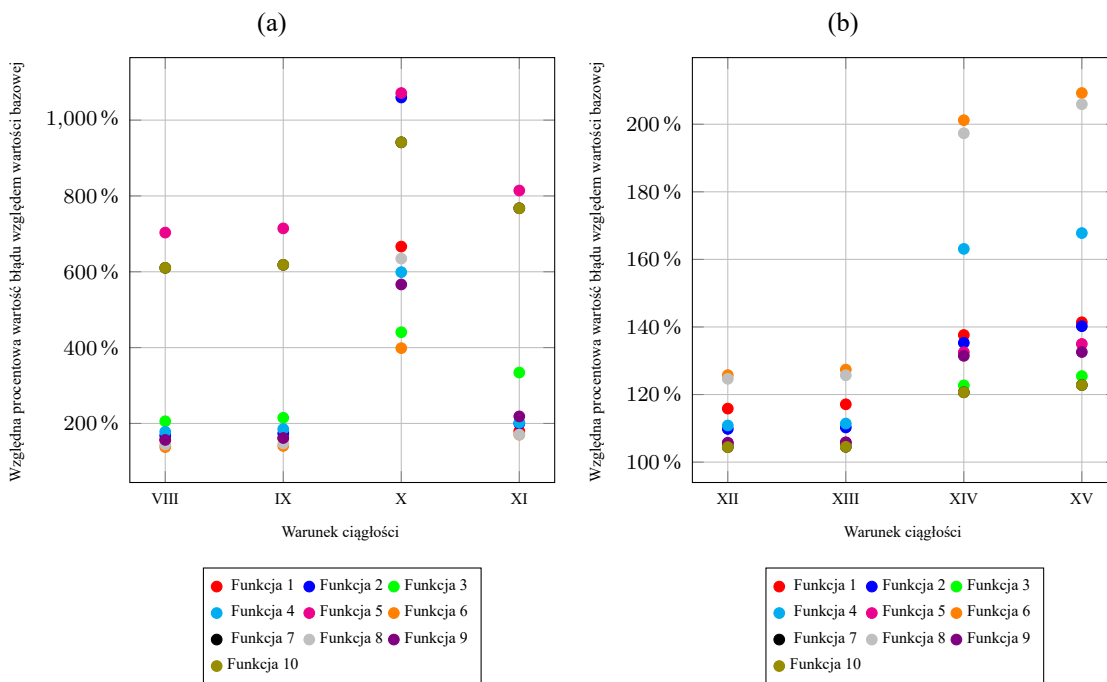
Wartość błędów dla bazowego przypadku 6.10 będzie poziomem do względem którego kolejne przypadki z tabeli (6.1) będą odnosić się w procentach. Wyniki analizy błędów dla prezentowanych przypadków przedstawiono na rysunkach 6.11 - 6.13.



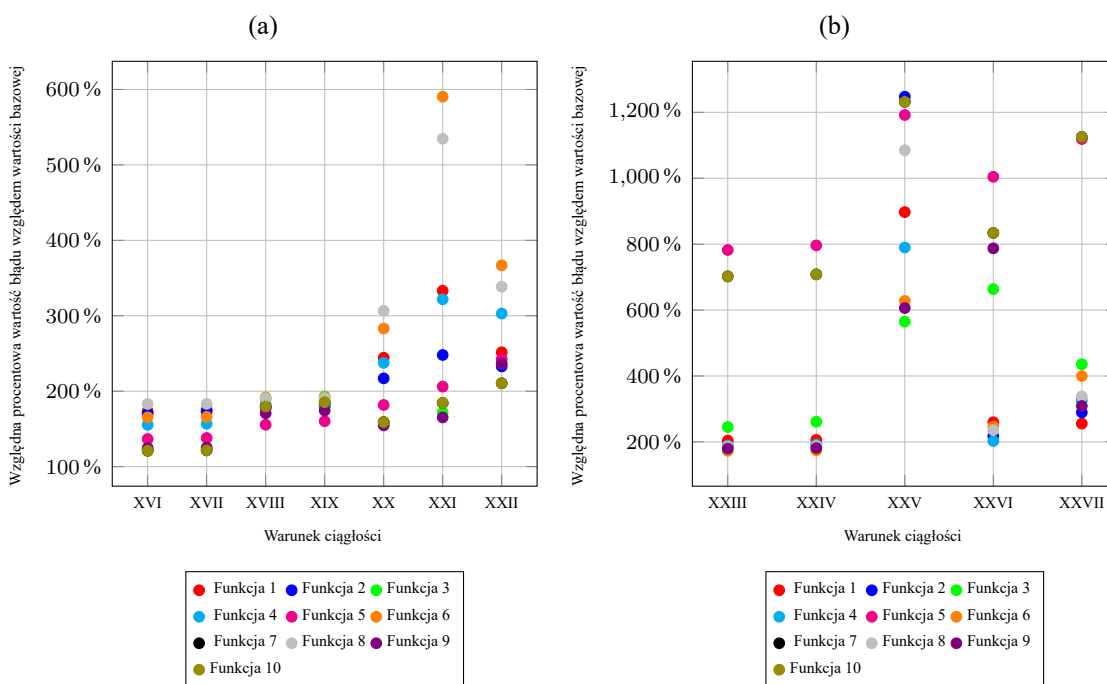
Rysunek 6.10: Wartości błędów aproksymacji względem normy $W^{1,2}$ testowych krzywych Béziera dla Warunku 1 z tabeli 6.1.



Rysunek 6.11: Wartości błędzi aproksymacji w normie $W^{1,2}$ dla testowych krzywych Béziera, względem wartości bazowej. (a) Warunek 2, (b) Warunek 3 z tabeli 6.1.



Rysunek 6.12: Wartości błędu aproksymacji w normie $W^{1,2}$ dla testowych krzywych Béziera, względem wartości bazowej. (a) Warunek 4, (b) Warunek 5 z tabeli 6.1.

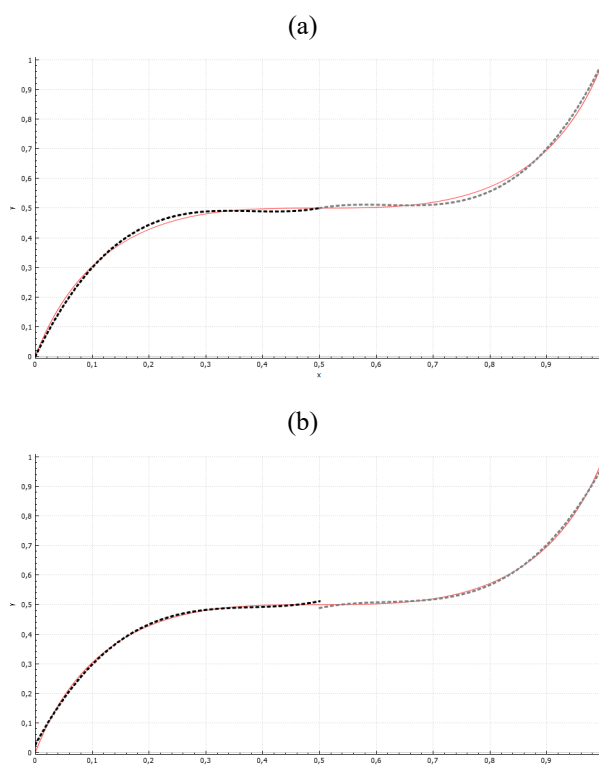


Rysunek 6.13: Wartości błędu aproksymacji w normie $W^{1,2}$ dla testowych krzywych Béziera, względem wartości bazowej. (a) Warunek 6, (b) Warunek 7 z tabeli 6.1.

Wartość bazowa błędu (dla warunku 6.10) jest najmniejsza względem prezentowanej normy, spośród wszystkich testowanych kombinacji warunków dodatkowych. Po dołączeniu dodatko-

wych warunków ciągłości w rozdziale 4.2, liczba stopni swobody dla współczynników wielomianów co najwyżej trzeciego stopnia ulega zmniejszeniu. Część tych współczynników nie jest już niezależna, lecz zależy od innych współczynników.

Można zauważyć, że wybranie normy $W^{1,2}$ do wyznaczenia funkcji minimalizującej wyznacza lepsze rezultaty niż według normy L^2 . Nawet brak narzucenia warunków na zachowanie ciągłości C^0 w węzłach aproksymacji powoduje, że wyznaczone wielomiany aproksymujące prawie zachowują tę ciągłość (*nieciągłość*, czyli skok dla testowych krzywych jest rzędu 0.0001) - w przeciwieństwie do normy L^2 rysunek 6.14.



Rysunek 6.14: Przedstawienie wyników aproksymacji, przy użyciu dwóch wielomianów aproksymujących z wykorzystaniem tylko funkcji minimalizacyjnej opartej o normę $W^{1,2}$ oraz normę L^2 . (a) Aproksymacja w normie $W^{1,2}$, (b) Aproksymacja w normie L^2 .

Fakt ten implikuje jeszcze jedną interesującą cechę: narzucenie koniecznych warunków ciągłości II lub interpolacji III generuje niezauważalny wzrost błędu aproksymacji w normie $W^{1,2}$ (wzrost wartości poniżej 10^{-6} i widoczny na rysunku (6.13)), natomiast wybranie jednego z warunków jest konieczne, aby aproksymacja mogła być wykorzystana w aplikacjach do produkcji animacji metodą klatek kluczowych (bez dokonywania dodatkowych zmian w tych aplikacjach).

Narzucenie warunków gładkości klasy C^1 oraz C^2 powoduje zwiększenie błędu aproksymacji dla testowych krzywych, dochodzące do 200%.

Prezentowana metoda dotyczy problemów aproksymacji tylko jednej krzywej Béziera trzeciego stopnia. Zachowanie ciągłości C^0 gwarantuje warunek konieczny (opisany w p.4.2.1), a

ewentualna gładkość klasy C^1 oraz C^2 z sąsiadującymi krzywymi może być zagwarantowana poprzez wymuszenie wartości pierwszej (lub pierwszej i drugiej) pochodnej wielomianu aproksymującego w węzłach zewnętrznych takich samych jak wartość analogicznych pochodnych funkcji wejściowej $C(x)$. Badania tych przypadków przedstawiają rysunki 6.12 (dla zachowania gładkości klasy C^1) oraz 6.13 (dla zachowania gładkości klasy C^2). Narzucenie tego typu ciągłości powoduje już względne zwiększenie błędu od ponad 120% (patrz Warunek 5) do ponad 1200% (patrz Warunek 7), gdzie samo narzucenie gładkości klasy C^2 znacznie podnosi wartość błędu aproksymacji.

Przedstawione wyniki sugerują, aby w praktycznych zastosowaniach prezentowanej metody wykorzystywać tylko konieczne warunki ciągłości. Jeżeli jednak odbiór animacji przez użytkownika jest niezadowolający (widoczny jest brak ciągłości w zachowaniu prędkości animacji), należy zastosować kombinację warunków ciągłości co najwyżej z gładkością klasy C_1 w węzłach wewnętrznych i/lub zewnętrznych.

6.4 Testy błędów aproksymacji ze względu na funkcję wewnętrzną i zewnętrzną

Ta część analizy metody prezentowanej w pracy została podzielona na dwa punkty, ponieważ jak zostało opisane w równaniu (3.6), wejściowa trajektoria $C(x)$ to złożenie dwóch funkcji: wielomianu co najwyżej trzeciego stopnia (funkcja f) oraz odwrotności co najwyżej trzeciego stopnia (funkcja g). Ponieważ odwrotność może zawierać funkcje przestępne (hiperboliczne lub trygonometryczne), więc funkcja $C(x)$ może być funkcją przestępną.

Punkt pierwszy: analiza wpływu funkcji wewnętrznej g na błąd aproksymacji, przy ustalonej postaci wielomianu zewnętrznej f ;

Punkt drugi: analiza wpływu wielomianu zewnętrznego f na błąd aproksymacji, przy ustalonej postaci funkcji wewnętrznej g .

Taka analiza pozwala określić, jak prezentowana metoda aproksymuje liczne postacie funkcji wejściowej $C(x)$, co równoważne jest analizą aproksymacji ze względu na postać funkcji $C(x)$ z rozdziału 3.1.3.

6.4.1 Testy dla błędów aproksymacji ze względu na funkcję wewnętrzną

Do analizy wykorzystano znormalizowaną krzywą Béziera o punktach kontrolnych:

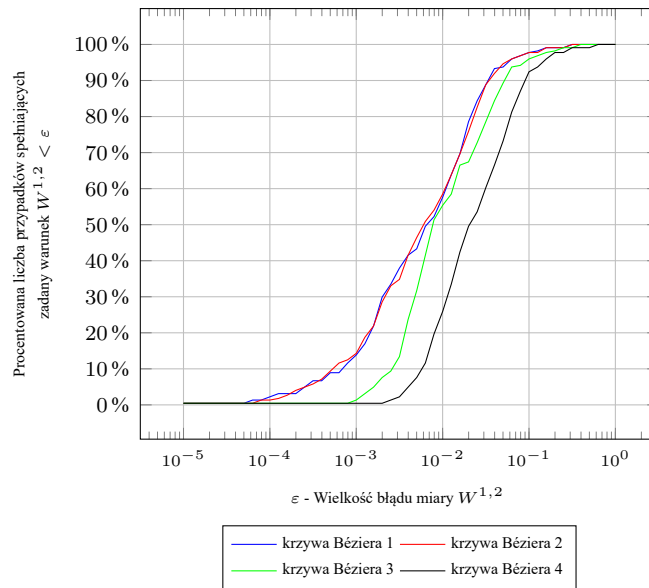
1. $P_0 = (0, 0)$, $P_1 = (a, \frac{1}{3})$, $P_2 = (b, \frac{2}{3})$, $P_3 = (1, 1)$ gdzie współrzędne y punktów kontrolnych generują funkcję zewnętrzną jako funkcję liniową.
2. $P_0 = (0, 0)$, $P_1 = (a, \frac{1}{6})$, $P_2 = (b, \frac{1}{2})$, $P_3 = (1, 1)$ gdzie współrzędne y punktów kontrolnych generują funkcję zewnętrzną jako funkcję kwadratową.

3. $P_0 = (0, 0), P_1 = (a, 0), P_2 = (b, 1), P_3 = (1, 1)$ gdzie współrzędne y punktów kontrolnych generują funkcję zewnętrzną która występuje jako wartość domyślna w implementacjach programów do animacji komputerowej.
4. $P_0 = (0, 0), P_1 = (a, -\frac{1}{2}), P_2 = (b, \frac{3}{2}), P_3 = (1, 1)$ gdzie współrzędne y punktów kontrolnych generują funkcję zewnętrzną która jest typu *EaseOutBack*.

Wykorzystano takie wartości stałej $N = 3$ oraz stałej $K = 20$ przy podziale równomiernym wzdłuż krzywej wejściowej $C(x)$.

Jak pokazano w rozdziale 3.1 punkty kontrolne znormalizowanej krzywej Béziera można przekształcić do analizowanej postaci (3.6), gdzie $a \in [0, 1] \wedge b \in [0, 1]$. Ten zakres ułatwi przedstawienie analizy w postaci graficznej. Przeanalizowane zostaną wszystkie rodziny krzywej $C(x)$ z rozdziału 3.1.3.

Dla lepszej wizualizacji wyników analizy, wyniki (czyli dane liczbowe) dodatkowo zostały pokolorowane z tą myślą, aby wartości minimalne (z określonego przedziału) przyjmowały odcienie koloru zielonego. A wartości maksymalne odcienie koloru czerwonego (zgodnie z opisem przedstawionym z prawej strony rysunków). Taka reprezentacja analiz określana będzie **mapą cieplną**.



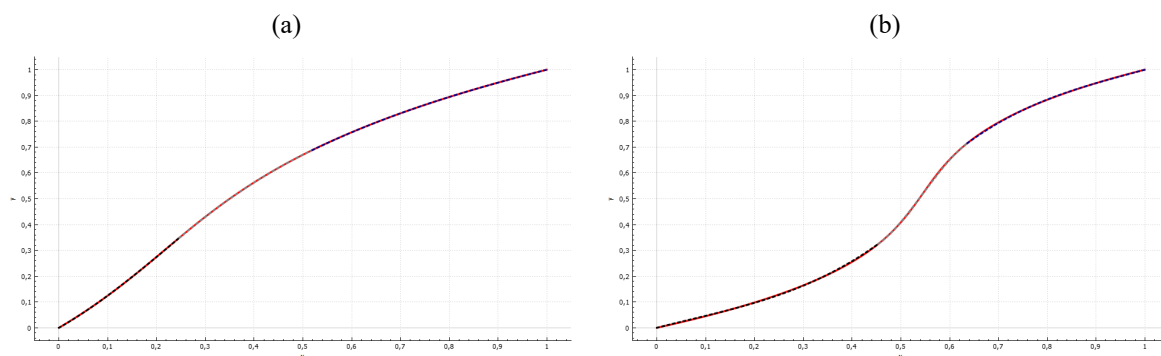
Rysunek 6.16: Procentowa liczba testowanych dla przykładowych krzywych Béziera których błąd aproksymacji jest mniejszy od zadanego.

Czerwoną czcionką na mapach cieplnych 6.15 zostały oznaczone błędy aproksymacji w normie L^2 dla funkcji z rodziny Ω_1 , a analiza tych przypadków została przedstawiona w p. 6.4.1.1.

Największe wartości błędów aproksymacji występują dla krzywych w których w przedziale wejściowym występują duże wartości pierwszej pochodnej ($C'(x) > 10^3$) Dla tak szybko zmieniających się wartości funkcji wejściowej, aproksymacja wielomianami co najwyżej trzeciego stopnia nie jest stanie dobrze naśladować funkcje wejściową $C(x)$.

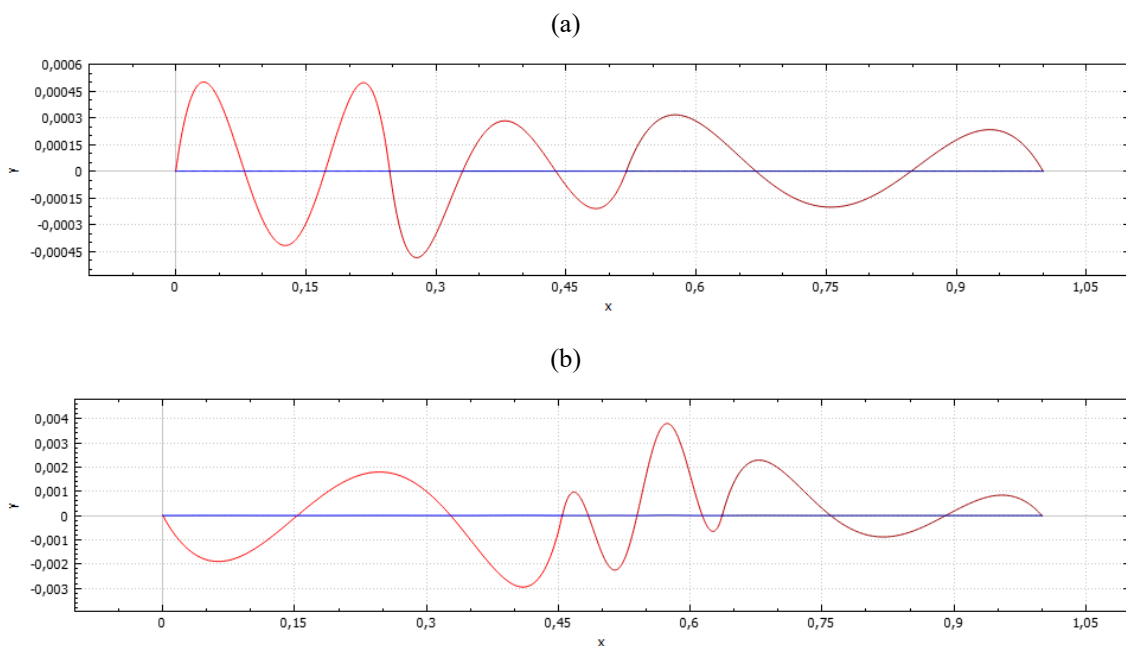
Z rysunku 6.16 można odczytać, że dla ponad 90% przypadków błąd aproksymacji jest mniejszy niż 0.1, a dzięki wcześniejszej analizie 6.1 można wnioskować, że największa różnica między funkcją wejściową a wielomianami aproksymującymi nie będzie większa niż 0.05 (średnio nie większa niż 0.01).

Pouczające jest przedstawienie funkcji wejściowej $C(x)$ i jej wielomianów aproksymujących we wspólnych wykresach, dla losowo wybranych przypadków.



Rysunek 6.17: Przykładowe krzywe Béziera (czerwona ciągła krzywa) oraz jej aproksymacja trzema wielomianami aproksymującymi (zaznaczona liniami przerywanymi). (a) Przykładowa krzywa Béziera 1, (b) Przykładowa krzywa Béziera 2.

Krzywą wejściową i jej aproksymację trudno rozróżnić na takim rysunku 6.17, dlatego do lepszej analizy można pokazać wykres (rysunek 6.18) różnicy między $C(x)$ a jej wielomianami aproksymującymi.



Rysunek 6.18: Różnice między przykładowymi krzywymi Béziera oraz jej trzema wielomianami aproksymującymi. (a) Przykładowa krzywa Béziera 1, (b) Przykładowa krzywa Béziera 2.

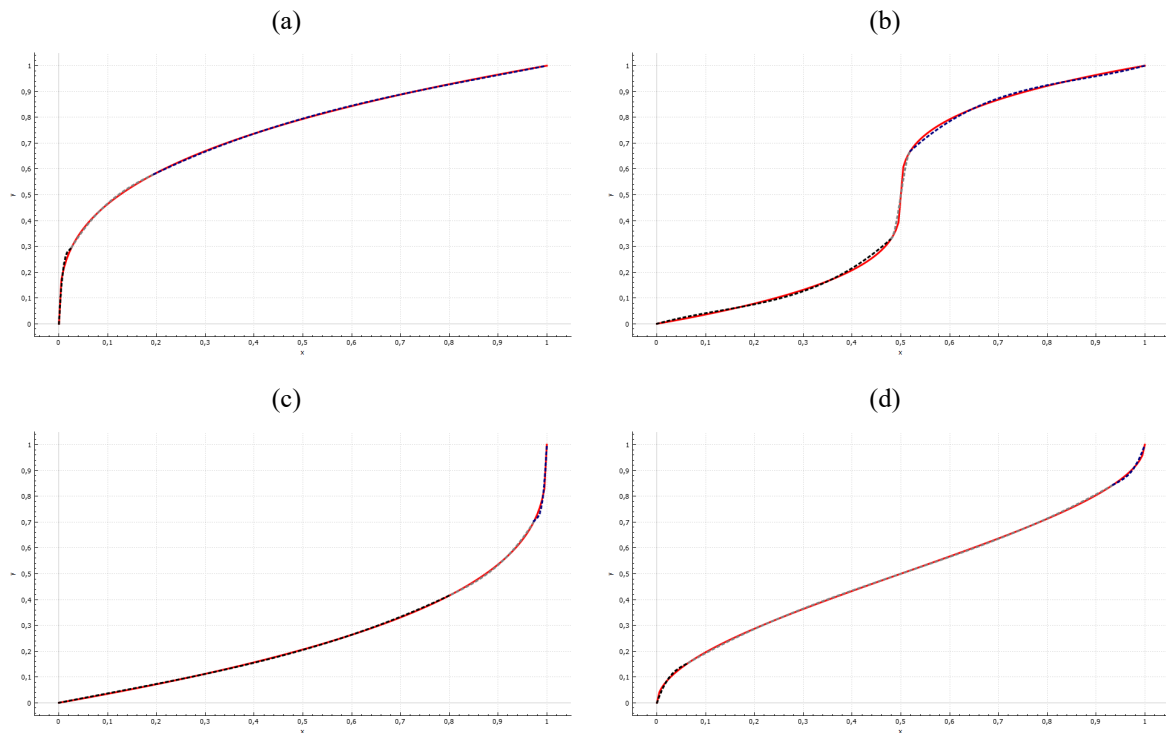
Wykres 6.18 potwierdza powszechny fakt w aproksymacji przy użyciu wielomianów, iż aproksymacja taka ma tendencje do oscylacji.

6.4.1.1 Analiza aproksymacji rodziny funkcji Ω_1

Funkcje rodziny Ω_1 charakteryzują się tym, że:

- Funkcja wewnętrzna g ma ekstremum lokalne na granicach przedziału $[x_0, x_1]$.
- Pochodna funkcji wewnętrznej g zeruje się w przedziale $x \in [x_0, x_1]$, ale nie następuje zmiana znaku pochodnej.

Dla funkcji wejściowej oznacza to, że jej pochodna ma asymptoty pionowe w przedziale $[x_0, x_1]$. Wejściowa funkcja w sąsiedztwie asymptoty posiada nieskończenie duże zmiany wartości. Pokazane zostało na przykładach trajektorii (rysunek 6.19).



Rysunek 6.19: Wykresy funkcji wejściowej $C(x)$ dla brzegowych przypadków oraz ich aproksymacje. (a) $a = 0 \wedge b = 0$, (b) $a = 1 \wedge b = 0$, (c) $a = 1 \wedge b = 1$, (d) $a = 0 \wedge b = 1$.

Wykresy wielomianów aproksymujących tylko w okolicy asymptoty pionowej pierwszej pochodnej funkcji wejściowej, widocznie różnią się od wykresu funkcji wejściowej co przedstawiono na rysunku 6.19.

Takiego typu trajektorie (z rodziny funkcji Ω_1) bardzo rzadko występują w praktycznych zastosowaniach, gdyż tak dużych zmian wartości funkcji nie wykorzystuje się często w animacjach. Ruch takiego obiektu bardzo szybko uciekałby z widocznego obszaru wirtualnej kamery, dodatkowo ruch ten wyświetlany w skończonej liczbie klatek generowanego obrazu traciłby iluzję płynności i sugerowałby, że taki obiekt porusza się skokowo.

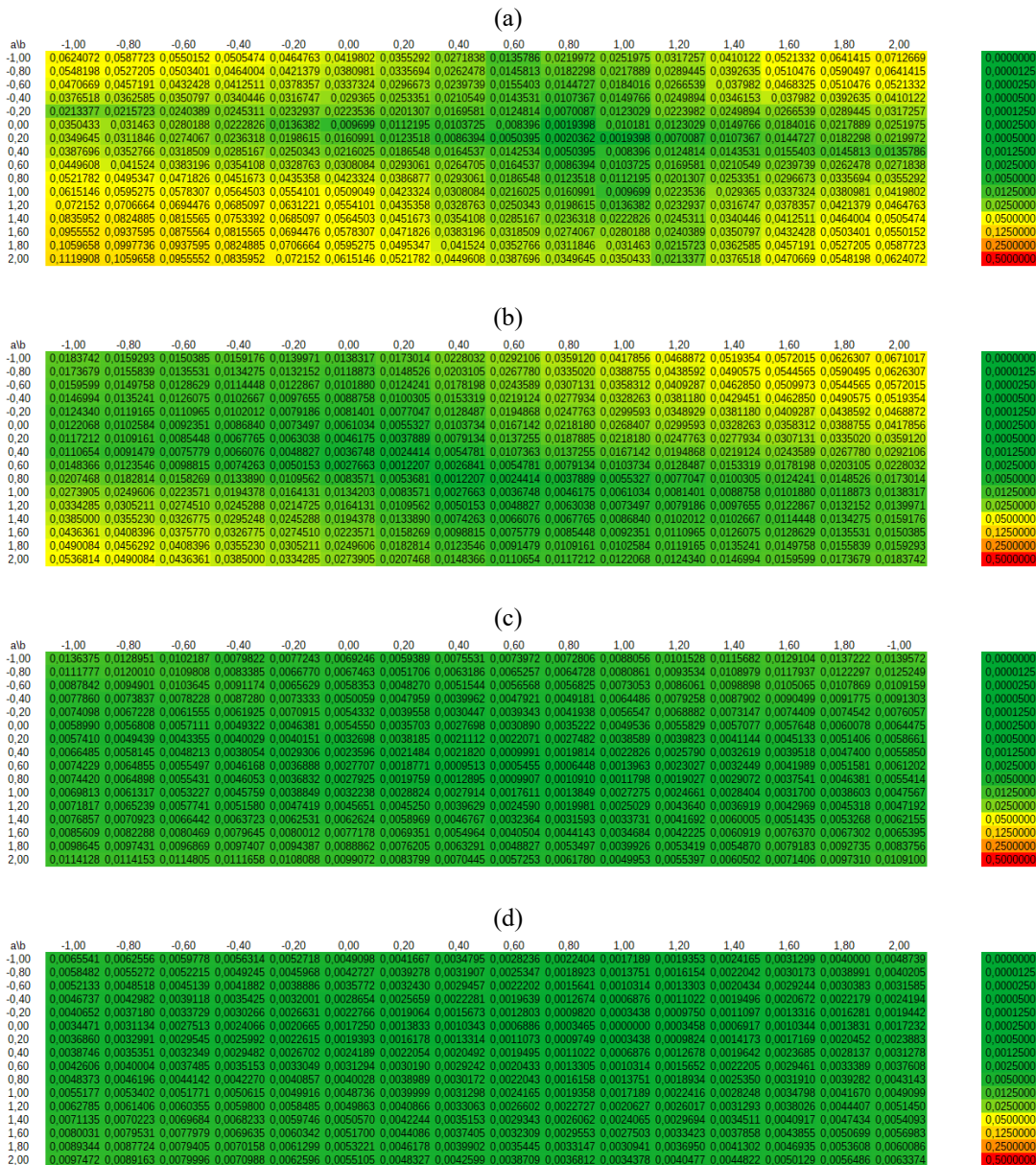
6.4.2 Testy dla błędów aproksymacji ze względu na funkcję zewnętrzną

Do analizy wykorzystano znormalizowaną krzywą Béziera o punktach kontrolnych:

1. $P_0 = (0, 0), P_1 = (\frac{1}{6}, a), P_2 = (\frac{5}{6}, b), P_3 = (1, 1)$, gdzie współrzędne x punktów kontrolnych generują funkcję zewnętrzną rodziny .
2. $P_0 = (0, 0), P_1 = (\frac{2}{3}, a), P_2 = (\frac{1}{3}, b), P_3 = (1, 1)$, gdzie współrzędne x punktów kontrolnych generują funkcję zewnętrzną rodziny .
3. $P_0 = (0, 0), P_1 = (\frac{2}{3}, a), P_2 = (\frac{2}{3}, b), P_3 = (1, 1)$, gdzie współrzędne x punktów kontrolnych generują funkcję zewnętrzną rodziny
4. $P_0 = (0, 0), P_1 = (0, a), P_2 = (1, b), P_3 = (1, 1)$, gdzie współrzędne x punktów kontrolnych generują funkcję zewnętrzną rodziny

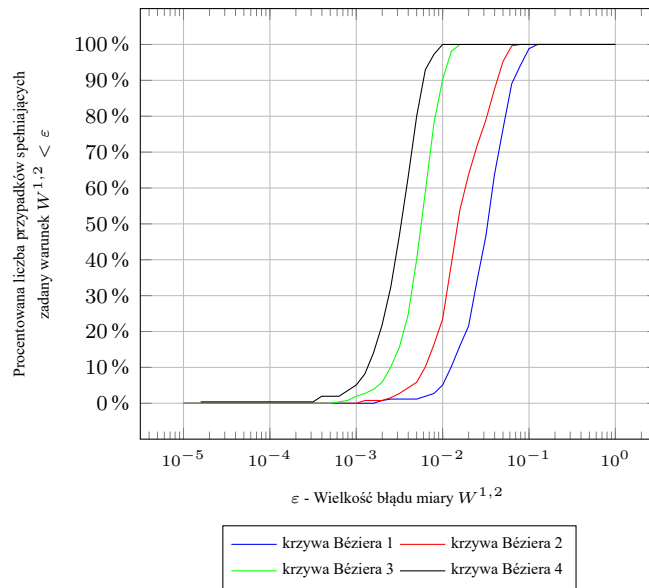
Wykorzystano takie wartości stałej $N = 3$ oraz stałej $K = 20$ przy podziale równomiernym wzdłuż krzywej wejściowej $C(x)$.

Dodatkowe zmienne mają zakres: $a \in [-1, 2] \wedge b \in [-1, 2]$. Taki zakres dla zmiennych a, b ułatwi przedstawienie analizy w postaci graficznej. Warto także wspomnieć, że gdy wartości zmiennych a oraz b , dla których odpowiadająca im funkcja zewnętrzna f ma w odcinku (x_0, x_3) ekstrema lokalne, w praktyce są rzadko wykorzystywane. Ponieważ prowadzą do budowy krzywych typu *EaseOutBack* co z kolei powoduje, iż z jednej strony animatorowi, który chciałby ich użyć, trudno jest oszacować wartość przestrzelenia (wynika to z faktu współrzędna y punktów kontrolnych punktów kontrolnych P_1 oraz P_2 nie przenoszą się w sposób liniowy na wartość przestrzelenia). Z drugiej strony samo przestrzelenie przeważnie jest zjawiskiem niepożądanym w interpolacji animowanego parametru.



Rysunek 6.20: Mapa ciepła błędów aproksymacji we względu na przypadki funkcji wewnętrznych. (a) Mapa ciepła dla krzywej Béziera 1, (b) Mapa ciepła dla krzywej Béziera 2, (c) Mapa ciepła dla krzywej Béziera 3, (d) Mapa ciepła dla krzywej Béziera 4.

Tak samo jak w testach funkcji wewnętrznej przedstawiony został także wykres procentowy na rysunku 6.21.



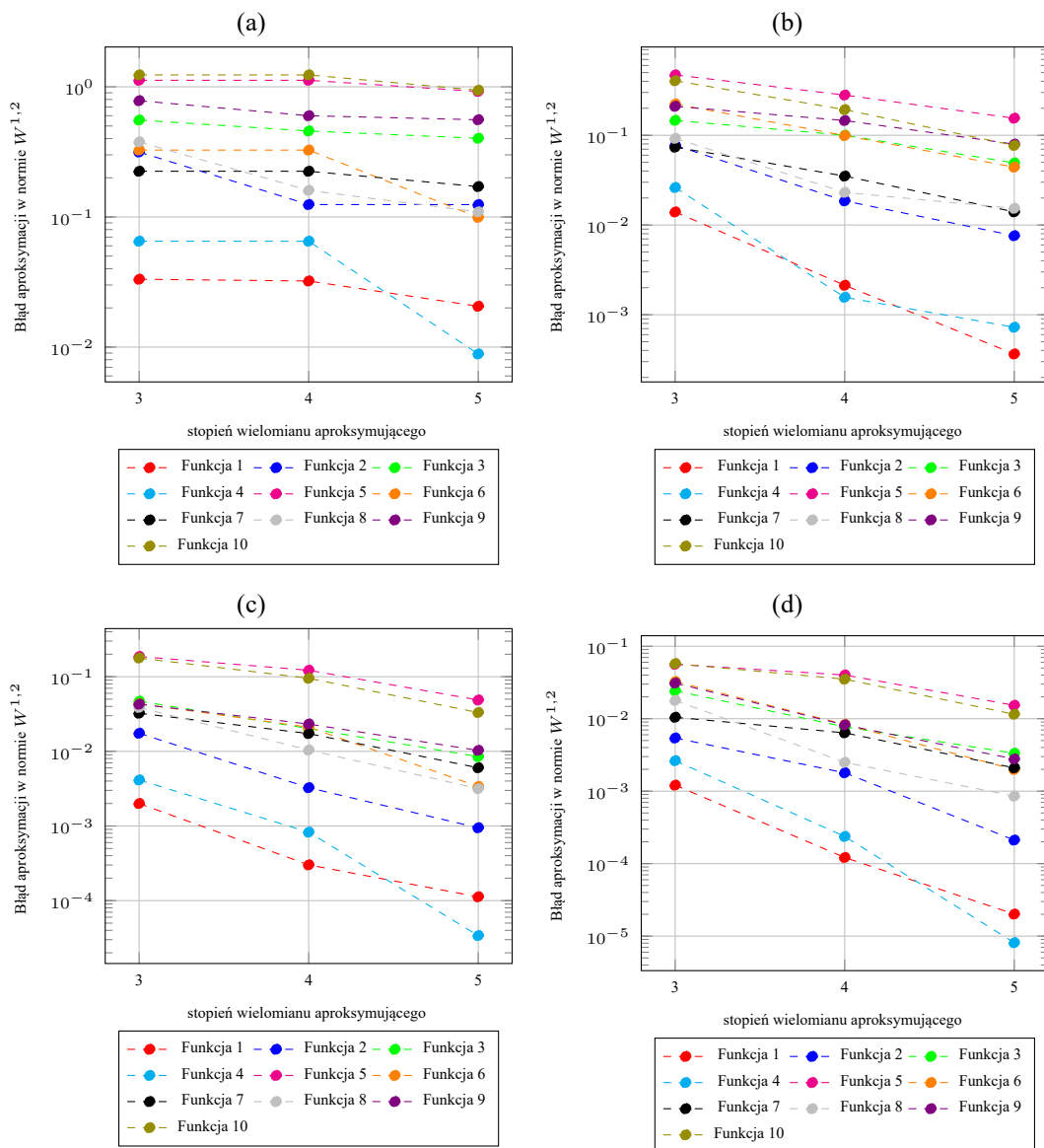
Rysunek 6.21: Krzywe procentowego rozkładu błędów przedstawionych na mapach cieplnych 6.20.

Gdy pochodna odwrotności funkcji wewnętrznej $(g(x)^{-1})'$ w rozpatrywanym przedziale $x \in [x_0, x_1]$ ma wartości mniejsze niż 1, wówczas funkcja zewnętrzna ma bardzo ograniczony wpływ na wartość błędów aproksymacji (w zakresie możliwych wartości współczynników podanym testom).

Gdy opisana wcześniej pochodna ma wartości $|(g(x)^{-1})'| > 2$ dla $x \in [x_0, x_1]$, wówczas funkcja zewnętrzna jest w stanie jeszcze bardziej wzmocnić przyspieszenia zmiany parametru i tym samym utrudnić dokładną aproksymację wielomianami co najwyżej trzeciego stopnia. Dla takich przypadków testowana aproksymacja trzema wielomianami aproksymującymi jest niewystarczająca (błąd aproksymacji jest za duży i aproksymowana trajektoria może być źle odbierana przez odbiorcę).

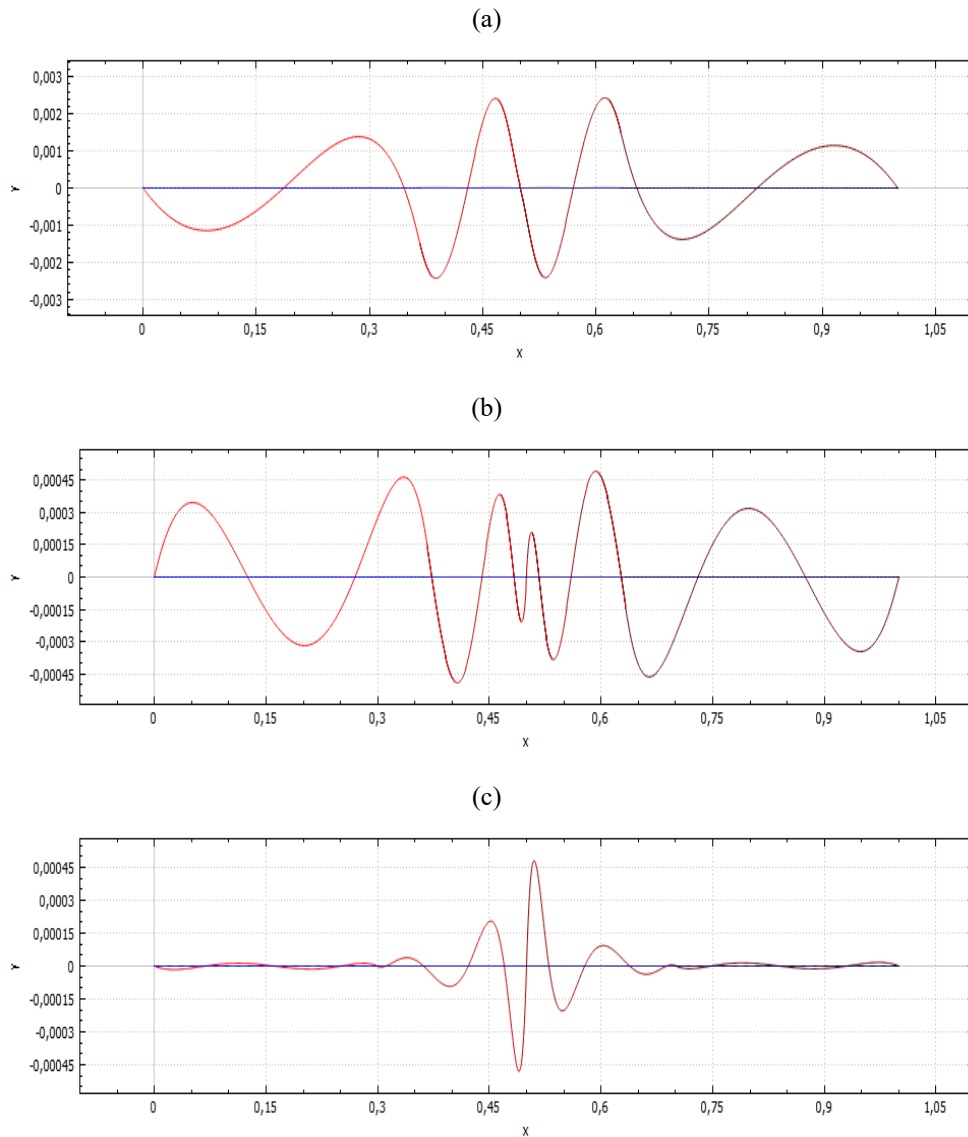
6.5 Rozszerzenie aproksymacji o wielomiany wyższych stopni

Jak wspomiano w rozdziale 4.6, prezentowaną metodę można wykorzystać do aproksymacji funkcji wejściowej $C(x)$ przy użyciu wielomianów wyższych stopni. Przebadano zatem, jak wielomiany czwartego oraz piątego stopnia wpływają na błąd aproksymacji. Dla różnej liczby wykorzystanych wielomianów w aproksymacji oraz stała $K = 20$ i przy podziale równomiernym wzdłuż krzywej wejściowej $C(x)$. Wyniki tych analiz przedstawione zostały na rysunku 6.22.



Rysunek 6.22: Wpływ stopnia wielomianów aproksymujących na błąd aproksymacji. (a) jeden wielomian aproksymujący, (b) dwa wielomiany aproksymujące, (c) trzy wielomiany aproksymujące, (d) cztery wielomiany aproksymujące.

Aproksymacja przy użyciu wielomianów wyższych stopni niż trzeci powoduje szybsze zmniejszenie się błędu aproksymacji w porównaniu do wielomianu co najwyżej trzeciego stopnia. Wynika to z faktu, że wielomiany wyższych stopni mogą oscylować wokół wejściowej trajektorii $C(x)$ z większą „częstotliwością”, co pokazane jest na rysunku (6.23). Wielomiany wyższych stopni niż trzeci mają niezerowe pochodne wyższych rzędów, podobnie funkcja wejściowa w większości jej możliwych postaci, przez co lepiej mogą aproksymować funkcje przestępne.



Rysunek 6.23: Oscylacje wielomianów wyższych stopni wokół aproksymowanej funkcji. (a) oscylacje przykładowego wielomianu co najwyżej trzeciego stopnia, (b) oscylacje przykładowego wielomianu co najwyżej czwartego stopnia, (c) oscylacje przykładowego wielomianu co najwyżej piątego stopnia.

Najmniejszy spadek błędu aproksymacji występuje przy użyciu tylko jednego wielomianu stopnia wyższego niż 3, w porównaniu do wielomianu co najwyżej trzeciego stopnia. Zwiększenie liczby wielomianów wyższych stopni, powoduje szybsze zmniejszenie błędu aproksymacji (nawet o jeden rząd wielkości). Dlatego, aby osiągnąć błąd akceptowalny, potrzebna jest mniejsza liczba wielomianów wyższych stopni w porównaniu do wielomianów co najwyżej trzeciego stopnia. Kosztem zwiększenia liczby rozkazów procesora typu FMA (z trzech dla wielomianie trzeciego stopnia do pięciu dla wielomianu piątego stopnia) podczas interpolacji wykorzystywanych aplikacji (co zostanie zbadane w rozdziale 6.9) oraz zwiększeniem zapotrzebowania pamięciowego na zapis współczynników z czterech dla wielomianu trzeciego stopnia do sze-

ściu dla wielomianu piątego stopnia. Ale to zwiększenie pamięciowe może być skompensowane przez fakt, iż w aproksymacji wykorzystana jest mniejsza liczba wielomianów (co zostanie zbadane w rozdziale 6.8).

Współczesne programy do animacji komputerowej, takie jak Blender, 3ds Max oraz Maya, nie mają możliwości interpolowania parametrów w metodzie klatek kluczowych przy użyciu wielomianów wyższych stopni, ponieważ interpolacja krzywymi Béziera w znikomym sposób wpływa na czas generowania pojedynczej klatki obrazu (np. filmu animowanego).

Silniki do produkcji gier wideo Unreal Engine 4 lub Unity także nie mają tej możliwości, ale zaadaptowanie tego sposobu w tym przypadku byłoby korzystne dla tych programów (z powodu omawianego w pracy płynącej korzyści minimalizacji problemów pamięciowo-wydajnościowych wobec stosowanych obecnych rozwiązań).

6.6 Porównanie wyników prezentowanej metody z wynikami interpolacji funkcjami sklejanymi

Analizując prezentowaną metodę można dojść do wniosku, że metoda ta jest rozszerzeniem metody interpolacji funkcjami sklejanymi trzeciego stopnia. Tak samo jak w interpolacji funkcjami sklejanymi trzeciego stopnia, danymi wejściowymi są węzły. Ale w odróżnieniu od interpolacji funkcjami sklejanymi do jednoznacznego wyznaczenia wielomianów nie potrzebne jest narzucanie warunków na węzły wewnętrzne (np. zachowanie gładkości klasy C^1 oraz C^2), jak i różnego typu warunki brzegowe (opisanych w dodatku D).

Oczywiście możliwy jest dobór warunków ciągłości w węzłach opisanych w rozdziale 4.2, które to warunki spowodują, że prezentowana metoda będzie równoważna z interpolacją funkcjami sklejanymi trzeciego stopnia różnych typów.

Przeprowadzono analizę, jak wartość błędu aproksymacji, uzyskana metodą prezentową w pracy z dwoma konfiguracjami warunków ciągłości w węzłach, odnosi się do błędu aproksymacji w normie $W^{1,2}$, obliczonego na podstawie wielomianów wygenerowanych przy użyciu interpolacji funkcjami sklejanymi (też w dwóch wariantach).

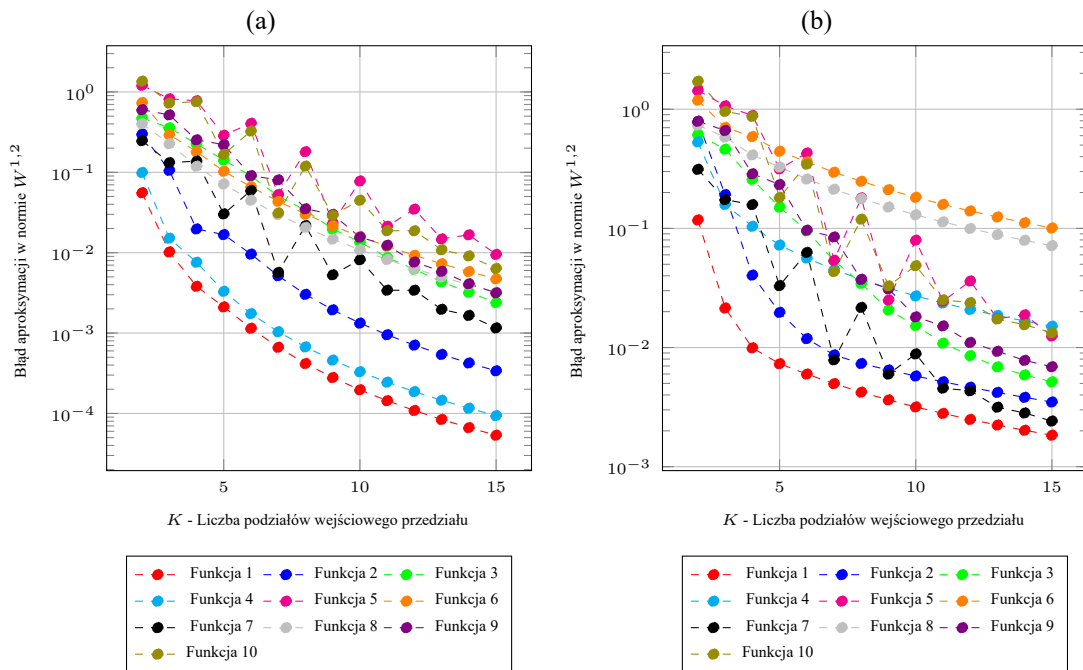
Testy zostały przeprowadzone na testowych krzywych dla różnej liczby wielomianów aproksymujących (warto zaznaczyć, że w tym przypadku zrezygnowano z metody siłowej).

1. Zaciśniętej funkcji sklejaney trzeciego stopnia (ang. *Clamped spline*).
2. Naturalnej funkcji sklejaney trzeciego stopnia (ang. *Natural spline*).
3. Wielomianem co najwyżej trzeciego stopnia z warunkami w węzłach zgodne z konfiguracją IX. Ta konfiguracja została wybrana z myślą, aby warunki w węzłach wewnętrznych były takie same jak w interpolacji krzywymi sklejanymi trzeciego stopnia. Brak w tej

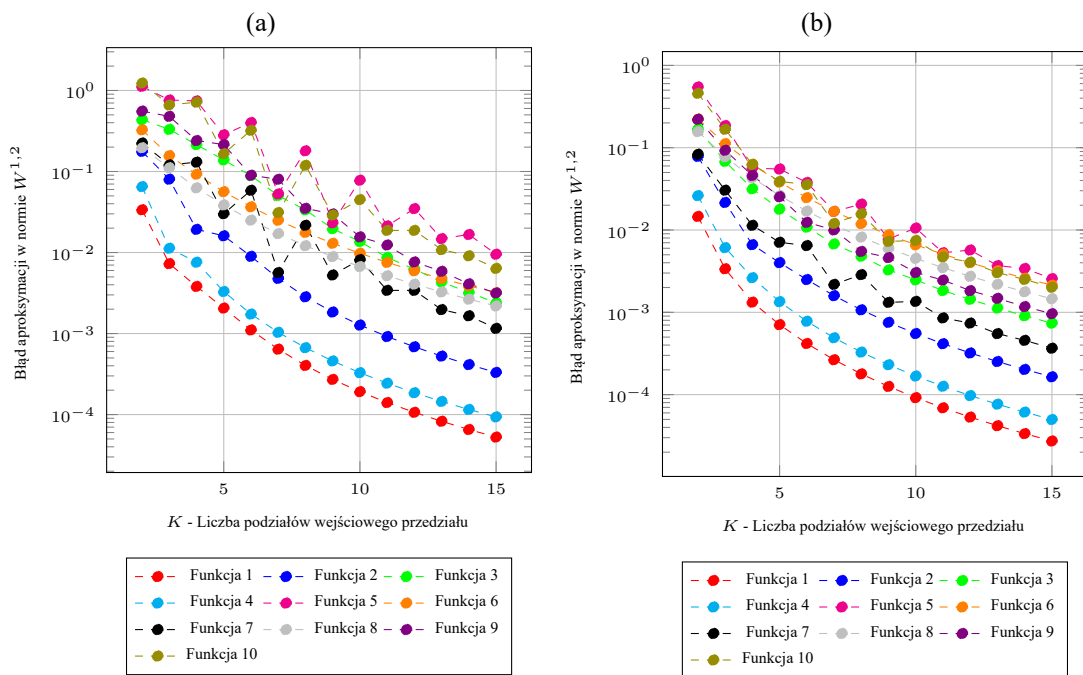
konfiguracji dodatkowych warunków w węzłach zewnętrznych. Te dodatkowe różnego rodzaju warunki powodują liczne rodzaje interpolacji funkcjami sklejanymi.

4. Wielomianem co najwyżej trzeciego stopnia z warunkami w węzłach zgodne z konfiguracją II. Ta konfiguracja została wybrana jako konieczne minimum warunków w prezentowanej pracy.

Wszystkie te testy podzielone zostały na dwa typy równomiernego podziału przedziału wejściowego (tak samo jak to miało miejsce w testach z rozdziału 6.2).

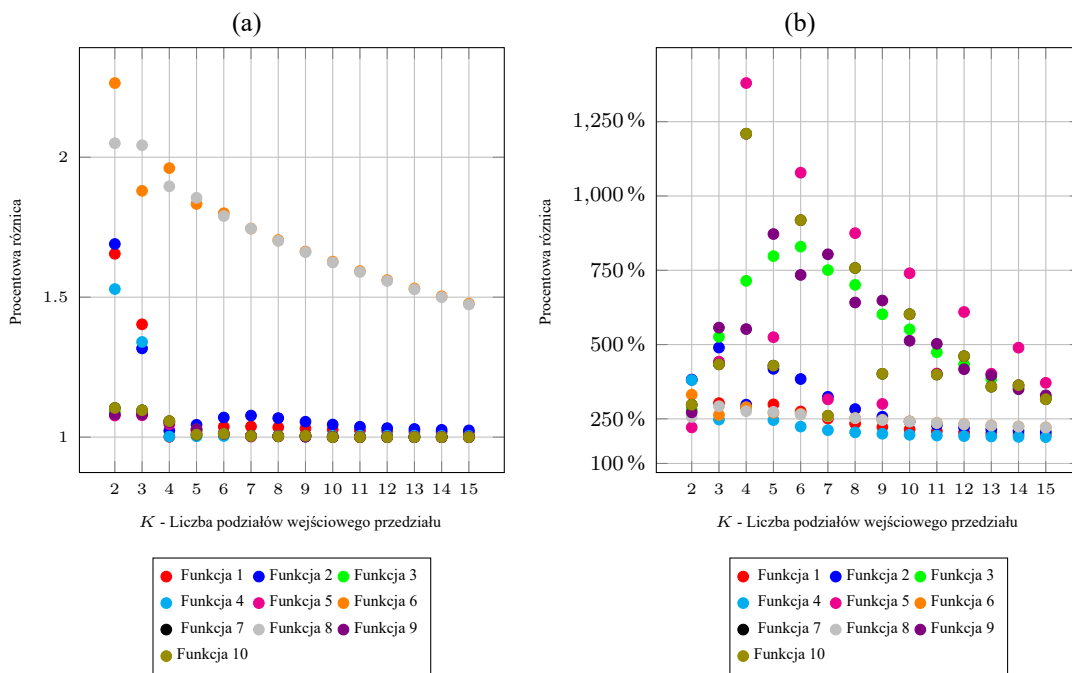


Rysunek 6.24: Błąd aproksymacji testowych krzywych przy użyciu funkcji sklejanych. (a) Zacięniętej funkcji sklejaney trzeciego stopnia, (b) naturalnej funkcji sklejaney trzeciego stopnia.



Rysunek 6.25: Błąd aproksymacji testowych krzywych przy użyciu wielomianu co najwyżej trzeciego stopnia. (a) Wielomian trzeciego stopnia z warunkiem IX, (b) wielomian trzeciego stopnia z warunkiem II.

Wykresy na rysunku 6.25, wyglądają dosyć podobnie i dobra ocena porównawcza metody funkcjami sklejanymi i metody prezentowanej w pracy jest utrudniona, zatem dodatkowo sprawdzono zostało jak wygląda stosunek błędu zaciśniętej funkcji sklejaney trzeciego stopnia do wielomianów trzeciego stopnia z warunkami IX oraz II.



Rysunek 6.26: Wykres stosunku błędu aproksymacji zaciśniętej funkcji sklejaney trzeciego stopnia do wielomianu trzeciego stopnia. Wartość 2 na wykresie oznacza to, że metoda prezentowana w pracy generuje dwukrotnie mniejszą wartość błędu aproksymacji. (a) Zaciśnięta funkcja sklejana trzeciego stopnia kontra wielomian trzeciego stopnia z warunkiem IX, (b) porównaniu zaciśniętej funkcji sklejaney trzeciego stopnia z wielomianem trzeciego stopnia z warunkiem II.

Z wykresów 6.26 można wywnioskować, że dodanie większej liczby stopni swobody w poszukiwaniu współczynników wielomianów aproksymujących (co ma miejsce w metodzie prezentowanej) powoduje, że aproksymacja funkcji wejściowej jest dokładniejsza, dla większości przypadków testowanych krzywych oraz liczby wielomianów aproksymujących.

Dodatkowe warunki zachowania ciągłości, z konfiguracji IX, ograniczają stopnie swobody w poszukiwaniu współczynników wielomianów aproksymujących do dwóch. Nawet przy tak ograniczonych stopniach swobody prezentowana metoda nie jest gorsza w interpolacji niż metodą funkcji sklejaneych. Dla mniejszej liczby wielomianów aproksymujących wyniki aproksymacji są lepsze w niewielkim stopniu. Istnieją jednak krzywe (testowa krzywa 6 oraz 8), dla których aproksymacja prezentowaną metodą prowadzi do błędu od 1.5 do 2 razy mniejszego.

Dla sugerowanej w pracy liczby warunków ograniczających się tylko do warunków koniecznych, błąd aproksymacji wielomianami co najwyżej trzeciego stopnia w porównaniu do funkcji sklejaneych trzeciego stopnia jest ponad dwa razy mniejszy (a w niektórych przypadkach nawet o rząd wielkości).

6.7 Postać znormalizowana a błąd aproksymacji

Elementem wartym przeanalizowania jest także sprawdzenie, jak błąd aproksymacji dowolnej krzywej Béziera metodą prezentowaną w pracy zmienia się w stosunku do błędu aproksymacji krzywej poddanej normalizacji opisanej w 5.1.1. W tym celu sprawdzone zostaną, na przykładzie krzywych testowych 5.1.3, przekształcenia afiniczne występujące w animacji komputerowej, które krzywe Béziera przekształcają do postaci znormalizowanej:

1. Skalowanie krzywej Béziera wzdłuż osi X

Wartość współrzędnych x punktów kontrolnych testowych krzywych pomnożone zostały przez testowe wartości, co równoważne jest wydłużeniu (skróceniu) czasu trwania interpolacji krzywa Béziera.

2. Skalowanie krzywej Béziera wzdłuż osi Y

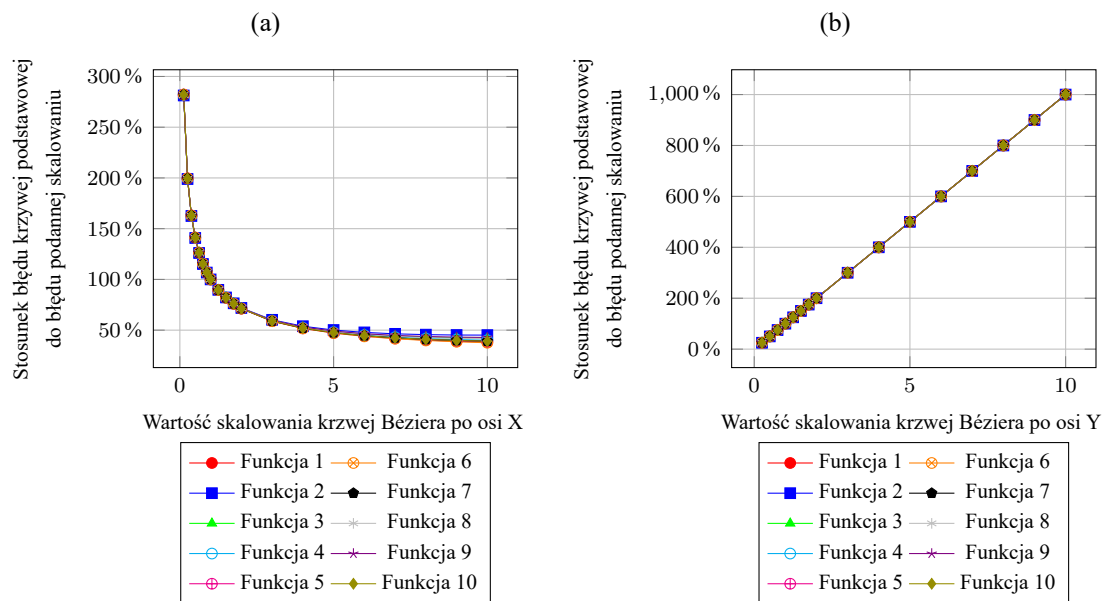
Wartość współrzędnych y punktów kontrolnych testowych krzywych pomnożone zostały przez testowe wartości, co powoduje zwiększenie wartości parametru, który podawany jest interpolacji.

3. Translacja krzywej Béziera wzdłuż osi X

Do wartości współrzędnych x punktów kontrolnych testowych krzywych dodane zostały testowe wartości. Wynikiem tej operacji jest przesunięcie w czasie danej interpolacji.

4. Translacja krzywej Béziera wzdłuż osi Y

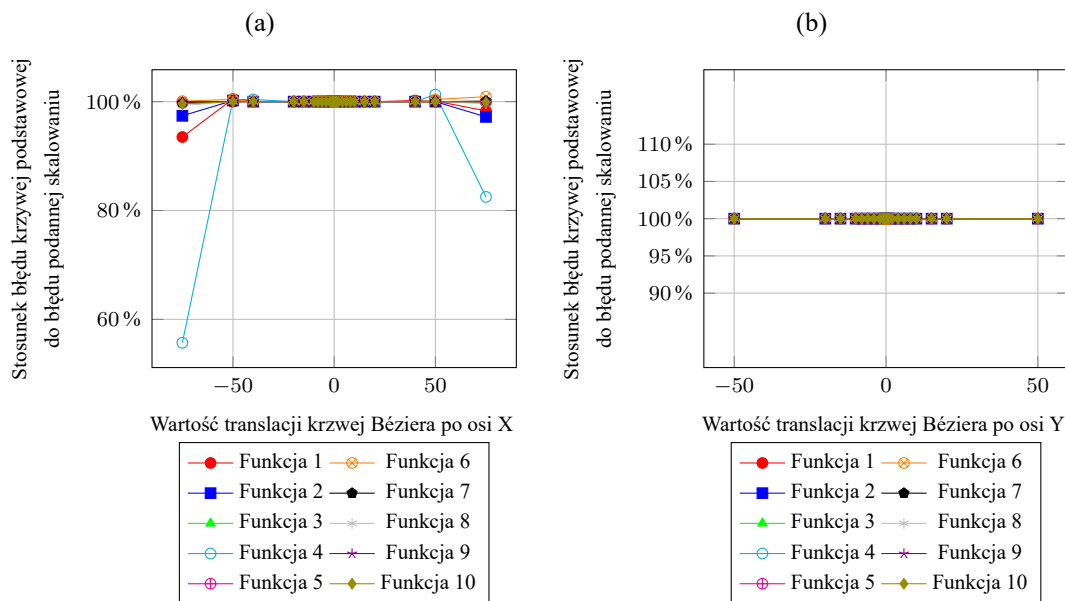
Do wartości współrzędnych y punktów kontrolnych testowych krzywych dodane zostały testowe wartości. Wynikiem tej operacji jest przesunięcie w początkowej i końcowej wartości parametru podawanemu animacji.



Rysunek 6.27: Wykresy wpływu skalowania znormalizowanych krzywych Béziera na błąd aproksymacyjny. (a) Skalowanie krzywej Béziera po osi X, (b) skalowanie krzywej Béziera po osi Y.

Oś Y krzywej Béziera wpływa na wzór funkcji zewnętrznej (3.7) i przy liniowym skalowaniu wartości parametrów błąd aproksymacji rośnie liniowo w stosunku do postaci znormalizowanej. W ogólnym przypadku wartość błędów może przyjąć dowolnie dużą wartość, ponieważ wartości współrzędnych y punktów kontrolnych P_0 oraz P_3 mogą przyjmować dowolną wartość (nie są one niczym ograniczone). Ale względny błąd względem wielkości $P_3y - P_0y$ (czyli różnica między początkową wartością parametru który poddawany jest animowaniu, a końcową jego wartością) procentowo jest taki sam jak dla postaci znormalizowanej krzywej Béziera. Gdy interpolowane są duże wartości animowanych parametrów, wartość błędów znormalizowanego pokrywa się z procentowym błędem wejściowych wartości.

Oś X krzywej Béziera wpływa na wzór funkcji wewnętrznej (3.7) i przy liniowym skalowaniu wartości parametrów błąd aproksymacji rośnie wykładniczo, przy ułamkowym wykładniku, w stosunku do postaci znormalizowanej. Skalowanie po wzdłuż X też ma wpływ na błąd, ale jej wpływ jest dużo mniejszy niż w poprzednim przypadku ze względu na charakter funkcji, która rośnie także do nieskończoności, ale dużo wolniej.



Rysunek 6.28: Wykresy wpływu translacji znormalizowanych krzywych Béziera na błąd aproksymacyjny. (a) translacja krzywej Béziera po osi X, (b) translacja krzywej Béziera po osi Y.

Translacja współrzędnych punktów kontrolnych krzywej Béziera wzdłuż osi X powoduje problemy z dokładnością obliczeniową liczb zmiennoprzecinkowych podwójnej precyzji, użytych w implementacji prezentowanej metody. W zakresie $\delta x \in (-40.0, 40.0)$ translacja nie ma wpływu na błąd aproksymacji.

Translacja współrzędnych punktów kontrolnych krzywej Béziera po osi Y nie ma wpływu wzrost błędu aproksymacji.

6.8 Testy pamięciowe

Teoretyczne zapotrzebowanie pamięciowe X dla różnego rodzaju interpolacji rozważanych w pracy opisują wzory (6.1)-(6.4). Przyjęto następujące oznaczenia: D to średni czas między klatkami kluczowymi, T to czas trwania animacji, K to liczba parametrów interpolowanych w animacji, f to częstotliwość próbkowania, S oznacza stopień wielomianu użyty w interpolacji. To miara zapotrzebowania na pamięć

- Interpolacja krzywą Béziera trzeciego stopnia

$$X = \left\lceil \frac{T}{D} \right\rceil * 8 * K \quad (6.1)$$

- Metoda próbkowania krzywej Béziera

$$X = T * f * K \quad (6.2)$$

- Funkcja wejściowa $C(x)$

$$X = \left\lceil \frac{T}{D} \right\rceil * K * 8 + \left(\left\lceil \frac{T}{D} \right\rceil + 1 \right) * K \quad (6.3)$$

- Interpolacja wielomianowa stopnia S

$$X = \left\lceil \frac{T}{D} \right\rceil * K * (S + 1) + \left(\left\lceil \frac{T}{D} \right\rceil + 1 \right) * K \quad (6.4)$$

Z wzorów (6.1)-(6.4) można wywnioskować, że zapis krzywych poddanych próbkowaniu nie jest zależny od liczby klatek kluczowych użytych w animacji, tylko od częstotliwości próbkowania. Dodatkowo nie jest potrzebne zapisywanie czasów dla tych próbek, tylko czas początku i końca animacji (pozostałe czasy można wyznaczyć na podstawie częstotliwości próbkowania i indeksu odtwarzanej próbki).

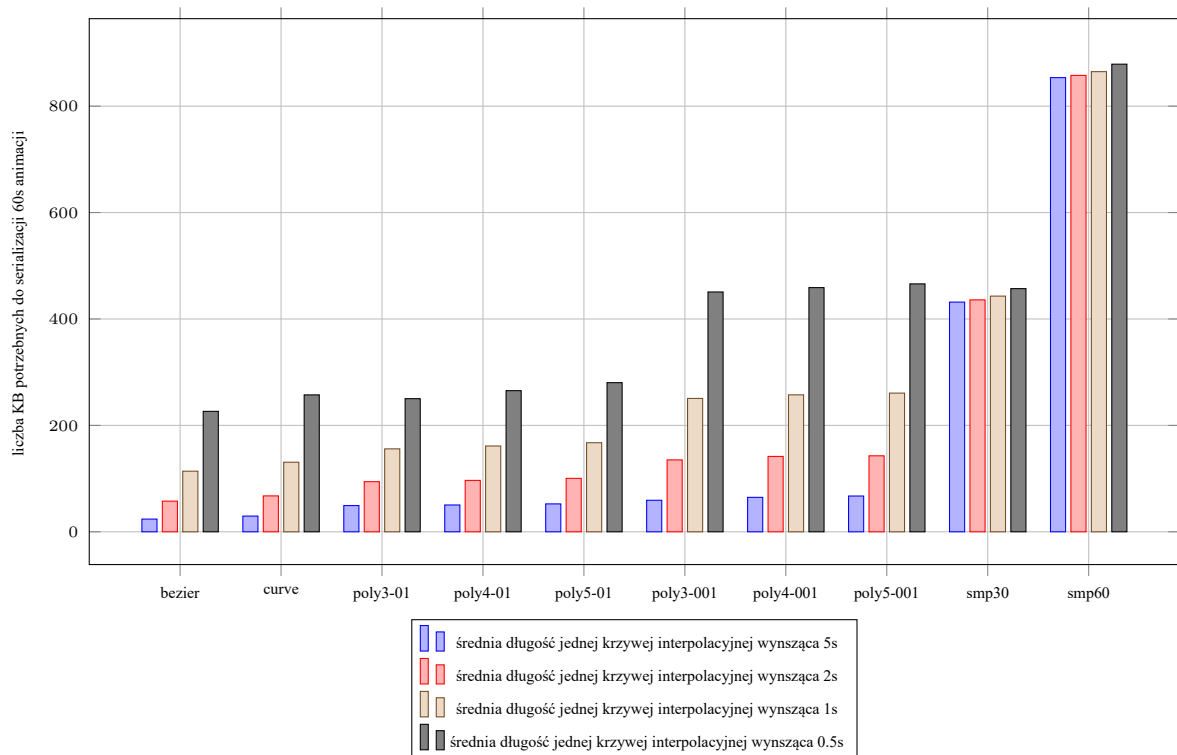
Interpolowanie krzywym Béziera trzeciego stopnia, wielomianami stopnia S oraz funkcją wejściową $C(x)$, dodatkowo potrzebne jest zapisywanie czasu animacji w którym dana krzywa będzie odtwarzana, albo te informacje są zawarte bezpośrednio w opisie interpolacji (jak ma to miejsce dla krzywej Béziera trzeciego stopnia, gdzie współrzędne x punktów kontrolnych P_0 oraz P_3 zawierają te informacje). Dla przypadku interpolacji wielomianami stopnia S oraz funkcji wejściowej $C(x)$ wartości te trzeba zapisać oddzielnie. Dodatkowo rozmiar potrzebny do zapisu tych danych zależy jest od średniego czasu trwania klatki kluczowej. Im więcej klatek kluczowych użytych jest w animacji parametrów, tym większe jest zapotrzebowanie na pamięć (które wzrasta liniowo).

Przeprowadzone zostały testy syntetyczne, aby sprawdzić, ile miejsca w pamięci potrzeba do zapisu jednej sekundy animacji składających się z losowo wybranych krzywych Béziera, o długości trwania jednej sekundy ($10 \text{ modeli} \times 60 \text{ kości na modelu} \times 6 \text{ parametrów}$: pozycja x, y, z w przestrzeni 3D, oraz trzy kąty Eulera). Taka konfiguracja kości oraz parametrów jest współcześnie wykorzystywana powszechnie w wysokobudżetowych grach komputerowych do reprezentacji oraz animacji postaci ludzkich np. w silniku Unreal Engine 4 [64].

W testach sprawdzono, jak sposoby zapisu funkcji interpolacyjnej wpływają na liczbę danych potrzebną do zapamiętania. Pod uwagę obrano następujące funkcje interpolujące:

- *bezier* - zapis punktów kontrolnych krzywej Béziera trzeciego stopnia.
- *curve* - zapis typu rodziny krzywej $C(x)$ oraz współczynników dla tej rodziny funkcji, oraz czasu podprzedziałów.
- *poly3-001* - zapis czterech współczynników wielomianu co najwyżej trzeciego stopnia oraz czasu podprzedziałów wyznaczonych z błędem aproksymacji funkcji $C(x)$ 0.0125.
- *poly3-01* - zapis czterech współczynników wielomianu co najwyżej trzeciego stopnia oraz czasu podprzedziałów wyznaczonych z błędem aproksymacji funkcji $C(x)$ 0.125.

- *poly4-001* - zapis pięciu współczynników wielomianu co najwyżej czwartego stopnia oraz czasu podprzedziałów wyznaczonych z błędem aproksymacji funkcji $C(x)$ 0.0125.
- *poly4-01* - zapis pięciu współczynników wielomianu co najwyżej czwartego stopnia oraz czasu podprzedziałów wyznaczonych z błędem aproksymacji funkcji $C(x)$ 0.125.
- *poly5-001* - zapis sześciu współczynników wielomianu co najwyżej piątego stopnia oraz czasu podprzedziałów wyznaczonych z błędem aproksymacji funkcji $C(x)$ 0.0125.
- *poly5-01* - zapis sześciu współczynników wielomianu co najwyżej piątego stopnia oraz czasu podprzedziałów wyznaczonych z błędem aproksymacji funkcji $C(x)$ 0.125.
- *smp30* - zapis wartości z próbkowanej funkcji $C(x)$ w częstotliwości 30Hz.
- *smp60* - zapis wartości z próbkowanej funkcji $C(x)$ w częstotliwości 60Hz.



Rysunek 6.29: Wykres zapotrzebowania pamięci na zapis 10s animacji składającej się z 360 parametrów.

Na rysunku 6.29 widoczne jest, że próbkowanie, które często jest wykorzystywane do zapisu animacji generuje bardzo dużą liczbę danych i liczba tych danych jest niezależna od długości krzywych interpolacyjnych z których składa się animacja.

Metoda próbkowania wymaga liczby danych ponad osiem razy więcej, przy 60Hz, oraz cztery razy więcej przy 30Hz, od liczby danych potrzebnych do zapisu punktów kontrolnych krzywej Béziera. Dlatego też bardzo często stosuje się różnego typu optymalizacje opisane w dodatku E.

Wielomiany aproksymujące trzeciego, czwartego i piątego stopnia zajmują tylko około 2 razy więcej miejsca. I aproksymacja wielomianami wyższych stopni, mimo że przy mniejszej liczbie wielomianów aproksymujących uzyskiwała wartość błędu mniejszą niż ustalona w teście 0.0125, nie powoduje, że liczba danych potrzebnych do zapisu współczynników jest mniejsza niż liczba danych potrzebnych do zapisania wielomianu trzeciego stopnia.

Zwiększenie dokładności aproksymacji metodą prezentowana w pracy powoduje zwiększenie liczby wielomianów koniecznych do aproksymacji co wiąże się ze zwiększeniem danych potrzebnych do zapisu animacji. W testowanym przypadku aproksymacja wielomianami do dokładności 0.0125 jest od 20% do 80% większa od aproksymacji o dokładności 0.125.

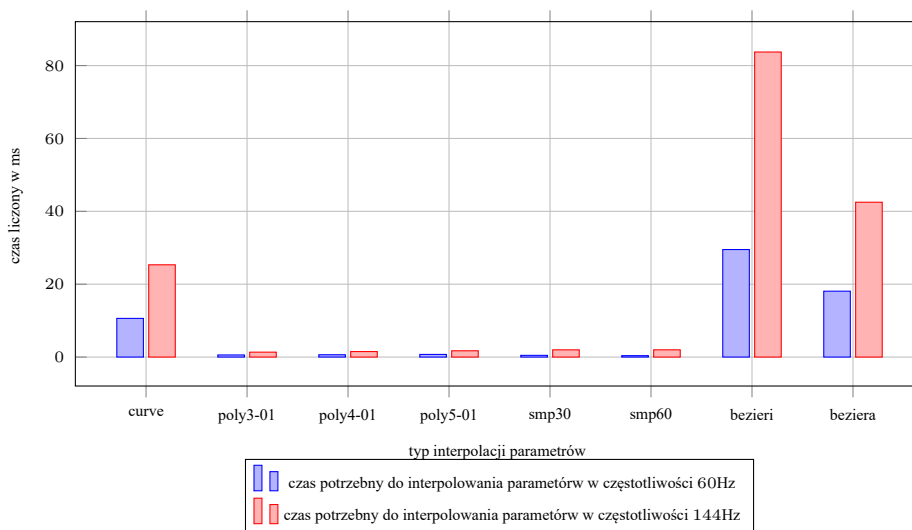
Zapis funkcji $C(x)$ potrzebuje tylko około 30% więcej miejsca, aby zapisać krzywą Béziera w parametryzacji (3.3).

6.9 Testy wydajnościowe

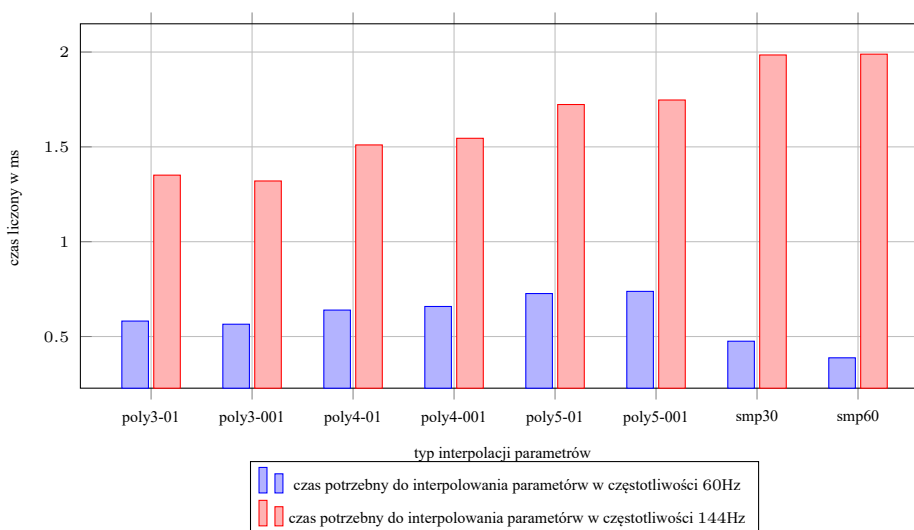
Testy wydajnościowe przeprowadzone zostały na jedenastu różnych sposobów wyznaczenia wartości funkcji interpolacyjnej:

- *curve* - interpolacja funkcji $C(x)$.
- *poly3-01* - interpolacja wielomianu co najwyżej trzeciego stopnia wyznaczona z błędem aproksymacji funkcji $C(x)$ 0.125.
- *poly3-001* - interpolacja wielomianu co najwyżej trzeciego stopnia wyznaczona z błędem aproksymacji funkcji $C(x)$ 0.0125.
- *poly4-01* - interpolacja wielomianu co najwyżej czwartego stopnia wyznaczona z błędem aproksymacji funkcji $C(x)$ 0.125.
- *poly4-001* - interpolacja wielomianu co najwyżej czwartego stopnia wyznaczona z błędem aproksymacji funkcji $C(x)$ 0.0125.
- *poly5-01* - interpolacja wielomianu co najwyżej piątego stopnia wyznaczona z błędem aproksymacji funkcji $C(x)$ 0.125.
- *poly5-001* - interpolacja wielomianu co najwyżej piątego stopnia wyznaczona z błędem aproksymacji funkcji $C(x)$ 0.0125.
- *smp30* - interpolacja liniowa danych próbkowanych z częstotliwością 30Hz.
- *smp60* - interpolacja liniowa danych próbkowanych z częstotliwością 60Hz.
- *bezieri* - interpolacja wykorzystująca algorytm wykorzystywany w programie Blender.
- *beziera* - interpolacja wykorzystująca algorytm numeryczny (znajdywania miejsca zerowego metodą Newtona-Raphsona [6]) przedstawiony w [48].

W tym celu przeprowadzono test, którego celem jest analizowanie czasu potrzebnego na interpolację dziesięciosiekundowej animacji trzystu sześćdziesięciu parametrów. Przeprowadzone zostały testy odtwarzane w czasie jednej klatki równej $\delta t = \frac{1}{60}$ s. Wartość ta jest powszechnie stosowana w grach komputerowych, gdyż jest równa wartości czasu odświeżania większości monitorów oraz telewizorów. Testy przeprowadzone zostały także dla $\delta t = \frac{1}{144}$ s, tę częstotliwość mają monitory wyższej klasy, oraz częstotliwości wyższe niż 60Hz stosowane są w okularach wirtualnej rzeczywistości (ang. *Virtual Reality*).



Rysunek 6.30: Wykres czasu interpolowania 10s animacji składającej się z 360 parametrów w zadanych częstotliwościach odtwarzania testowanych typów interpolacji.



Rysunek 6.31: Wykres czasu interpolowania 10s animacji składającej się z 360 parametrów w zadanych częstotliwościach odtwarzania wybranych typów interpolacji.

Rysunek 6.31 przedstawia wycinek testowanych interpolacji z rysunku 6.30, aby w ten sposób lepiej uwidocznic ich wydajność.

Prezentowana w pracy metoda wydajnościowo porównywalna jest do metody próbkowania. Metoda próbkowania (z częstotliwością 60Hz) jest szybsza w czasie odtwarzania animacji z częstotliwością 60Hz, ponieważ ta częstotliwość jest równa częstotliwości próbkowania, przez co interpolacja nie jest konieczna w ogóle (dane interpolacji pobierane są bezpośrednio z tablicy).

Natomiast, gdy częstotliwość odtwarzania nie jest wielokrotnością częstotliwościową próbkowania, wyznaczenie wartości interpolacji wymaga dodatkowej liniowej interpolacji między próbkami, co powoduje ponad czterokrotny wzrost czasu potrzebnego do uzyskania wartości parametru w zadanym czasie.

Częstotliwość odtwarzania nie wpływa negatywnie na metody interpolacyjne. Interpolacja wielomianami trzeciego stopnia przebiega wolniej przy częstotliwości 60Hz o około 80% w porównaniu z metodami próbkowania, w 144Hz jest od nich o 25% szybsza.

Na rysunku 6.30 widoczne jest, że czas potrzebny na interpolowanie metodami *curve* (czyli wyznaczanie wartości bezpośrednio z funkcji $C(x)$) oraz metody stosowane w aplikacjach, takich jak Blender, jest minimum rząd wielkości większy niż interpolacja wielomianami lub też metody próbkowania. Dlatego też użycie tych metod w aplikacjach czasu rzeczywistego powoduje duży narzut obliczeniowy i możliwe jest jej stosowanie, gdy liczba parametrów używanych w aplikacji jest niewielka (wyznaczana dla konkretnego przypadku aplikacji).

Rozdział 7

Podsumowanie

Niniejsza praca obejmuje zagadnienia związane z opisem trajektorii ruchu w animacji komputerowej. Rozpatrywany problem dotyczy nowego sposobu interpolacji parametrów w metodzie klatek kluczowych dla aplikacji czasu rzeczywistego. Do interpolacji zastosowana została krzywa Béziera trzeciego stopnia. W rozdziale 2 został przedstawiony cel oraz została sformułowana teza pracy. Przedstawiono także zadania potrzebne do osiągnięcia celu – zadania, których realizacja jest potrzebna do wykazania słuszności postawionej tezy. Wszystkie zaplanowane zadania zostały pomyślnie zrealizowane.

Krzywe Béziera trzeciego stopnia są bardzo często wykorzystywane w grafice komputerowej oraz animacji. Parametryzacja, w jakiej taka krzywa jest najczęściej przedstawiana, nie daje możliwości efektywnego wykorzystania jej jako funkcji interpolującej w metodzie klatek kluczowych aplikacjach czasu rzeczywistego. Najczęściej prezentowana parametryzacja tej krzywej nie jest funkcją czasu, co jest potrzebne do wyznaczania animowanych parametrów pomiędzy klatkami kluczowymi. Algorytmy stosowane w grafice komputerowej wykorzystują obliczeniowo kosztowny, dwuetapowy algorytm, który pozwala wyznaczyć wartość funkcji interpolującej (lub jej numeryczne przybliżenie).

Problem ten można rozwiązać za pomocą zmiany parametryzacji. Takie rozwiązanie zostało zaproponowane w pracy. Jednocześnie, aby zapewnić możliwość wykorzystania funkcji interpolacyjnej w metodzie klatek kluczowych, musi być spełniony warunek, aby krzywa interpolująca nie zawiera łuków albo pętli. Analiza tego warunku wykazała konieczność wprowadzenia ograniczeń na punkty kontrolne krzywej Béziera trzeciego stopnia. Umieszczenie punktów kontrolnych w ogólności może prowadzić do powstawania łuków lub pętli. W interpretacji krzywej Béziera jako krzywej interpolacyjnej skutkowałoby to koniecznością lokalnego „cofanie czasu” podczas animowania parametrów. Funkcja interpolująca musi być funkcją ściśle rosnącą lub ewentualnie niemalejącą przy zachowaniu tylko jednego punktu stacjonarnego w przedziale interpolacji. Zadanie odpowiednich warunków na punkty kontrolne krzywej Béziera gwarantuje, że funkcja interpolująca będzie funkcją niemalejącą.

Warunki ciągłości zostały podzielone na dwie grupy. W pierwszej znalazły się warunki konieczne, które są niezbędne dla zachowania ciągłości klasy C^0 . Zarówno w węzłach wewnętrznych, jak i w węzłach zewnętrznych. Zachowanie ciągłości klasy C^0 jest niezbędne także z czysto praktycznego powodu. Pozwala zapewnić możliwość zapisu zbioru uzyskanych wielomianów aproksymacyjnych w plikach wymiany danych współczesnych programów graficznych. Pozwoli to na odtwarzanie animacji (z wykorzystaniem zaproponowanej metody) w programach graficznych bez konieczności dokonywania dodatkowej modyfikacji.

W drugiej grupie znalazły się warunki opcjonalne. Zaproponowana metoda jest na tyle elastyczna, że pozwala użytkownikowi określić wybrane właściwości trajektorii ruchu definiując dodatkowe, opcjonalne warunki ciągłości wielomianów w węzłach wewnętrznych oraz zewnętrznych.

Jednym z trudniejszych problemów, przed jakimi stanął autor, była minimalizacja pozwalająca wyznaczyć jednoznacznie brakujące współczynniki wielomianów trzeciego stopnia dla aproksymacji krzywej Béziera. Aby rozwiązać to zadanie, konieczny był wybór odpowiedniej metryki, w której istnieje ekstremala dla zadanego problemu optymalizacji. Przy czym poszukiwana była metryka, w której możliwe byłoby wyznaczenie ekstremali bez złożonych metod obliczeniowych lub zaawansowanych metod numerycznych.

Teoretycznie najlepszą normą dla rozpatrywanego zadania wydaje się być norma supremum. Zbieżność w normie supremum (zbieżność jednostajna) implikuje zbieżność punktową. Minimalizacja normy supremum dałaby najlepszą miarę poprawności aproksymacji w rozpatrywanym problemie opisu ruchu w animacji. Najlepsza miara w tym przypadku oznaczałaby najłatwiejszy sposób interpretacji obliczonej wielkości będącej dokładnością aproksymacji. Wartość ta, w tym przypadku wartość normy supremum, wyraża największą bezwzględną różnicę między trajektoriami. W przeciwieństwie do norm całkowitych, dokładność w tej normie związana jest z percepcją ruchu – co jest decydującym czynnikiem wpływającym na odbiór animacji przez widza. Niestety wyznaczenie najlepszej aproksymacji w sensie normy supremum byłoby zadaniem trudnym. Wymagałoby obliczenia bezwzględnej różnicy pomiędzy wartościami funkcji w każdym punkcie przedziału ich określoności. W rozpatrywanych w pracy przypadkach, mogłoby to prowadzić do równań nierozwiązywalnych metodami algebraicznymi. Z kolei zastosowanie metod przybliżonych wiązałoby się z utratą kontroli nad rzeczywistym błędem aproksymacji lub wydłużeniem obliczeń poza możliwości realizacji w czasie rzeczywistym.

Aby rozwiązać problemy metryki, zaproponowano wykorzystanie przestrzeni Sobolewa. Norma w przestrzeni Sobolewa dostarcza górnego oszacowania normy supremum. Normę Sobolewa w przestrzeni $W^{1,2}$ wykorzystano do wyznaczenia minimalizatora aproksymacji funkcji wejściowej $C(x)$, ponieważ daje ona możliwość oszacowania błędu aproksymacji liczonego w supremum. Przy czym w zaproponowanym w pracy rozwiązaniu norma $W^{1,2}$ jest stosowana tylko i wyłącznie do funkcji, dla których możliwe jest wyznaczenie całek w całym przedziale

wejściowym. Dla pozostałych funkcji jest stosowana norma L^2 .

Zaproponowano metodę jednoznacznego wyznaczenia współczynników wielomianów bazującą na metodzie mnożników Lagrange’a. Ponieważ zastosowanie tylko warunków koniecznych oraz opcjonalnych (w większości przypadków kombinacji) nie daje możliwości jednoznacznego wyznaczenia współczynników wielomianów aproksymujących, wykorzystano ekstremum funkcji minimalizującej (dla przypadków funkcji brzegowych), leżące na powierzchni opisanej przez warunki konieczne i opcjonalne. Problem taki jest znany w teorii optymalizacji jako problem wyznaczenia ekstremum warunkowego funkcji różniczkowalnej. Metoda mnożników Lagrange’a jest jedną z podstawowych metod rozwiązania tego problemu.

Zaproponowana metoda wykorzystująca nową parametryzację krzywej Béziera trzeciego stopnia spełnia założenia funkcjonalne rozważane w celu pracy. Jednocześnie wykorzystanie minimalizacji funkcji opartej na mierze w przestrzeni Sobolewa pozwala wyznaczyć szukane współczynniki zgodnie z przyjętymi założeniami. Aby sprawdzić poprawność uzyskanych efektów interpolacji została przeprowadzona analiza rozwiązania. Praktycznie nie jest możliwa analiza wszystkich możliwych przypadków, ani ich klasyfikacja pozwalająca odrzucić (lub potwierdzić poprawność) części z nich. Wyznaczono więc postać znormalizowaną krzywej Béziera trzeciego stopnia, ułatwiającą analizę i przeprowadzenie testów zaproponowanej metody. Jednak żadna prosta miara matematyczna nie pozwala ocenić błędów aproksymacji trajektorii ruchu w animacji bez uwzględnienia percepcji człowieka. To widz decyduje o akceptacji oglądanej sekwencji ruchu. Zaproponowano szereg testów, aby możliwa była wizualna ocena jakości aproksymacji. Analiza ta pokazała, że istnieje błąd graniczny aproksymacji akceptowalny wizualnie. Jednak, aby można było oszacować jego wartość, potrzebne są szerokie statystyczne badania percepcyjne znacznie wykraczające poza zakres niniejszej pracy. Warto natomiast podkreślić możliwość efektywnej kontroli tego błędu w zaproponowanym rozwiązaniu. Ponieważ zbieżność ciągu funkcyjnego w normie Sobolewa $W^{1,2}$ na odcinku (a, b) implikuje zbieżność jednostajną, więc błąd aproksymacji mierzony w normie Sobolewa ogranicza z góry błąd aproksymacji w normie supremum z dokładnością do stałej $|a - b|$. A zatem zadając z góry odpowiednią wielkość błędu w normie Sobolewa, można uzyskać pożądaną wielkość błędu w normie supremum. Dokonując dostatecznie drobnego podziału odcinka (a, b) , można uzyskać odpowiednio małą wartość błędu w normie supremum, aby zapewnić poprawność percepcyjną i akceptację widza.

Wykorzystując postać znormalizowaną krzywej Béziera, zaprojektowano zestaw testów. Zostało wygenerowanych ponad 7 mln przypadków obejmujących pełny zakres zmienności parametrów znormalizowanej krzywej do lepszego górnego oszacowania błędu w normie Sobolewa względem normy supremum. Analiza wskazała że błąd w normie Sobolewa jest w najgorszym przypadku tylko dwukrotnie większy niż błąd w normie supremum, a w najlepszym dwudziestokrotnie większy.

Analizy wykazały, że podczas zwiększania liczby użytych wielomianów do aproksymacji błąd ten maleje co powoduje, że aproksymacja staje się dokładniejsza. Metoda brutalnej siły pozwala wyznaczyć przedziały o niejednakowych długościach, w których aproksymacja jest dokładniejsza niż równomierny podział wejściowego przedziału wykorzystując tę samą liczbę wielomianów aproksymujących. Podział wzdłuż krzywej wejściowej pozwala uzyskiwać lepszą aproksymację w porównaniu z podziałem zmiennej czasowej x .

Proponowana metoda dla aproksymacji wykorzystującej tylko trzy wielomiany, dla ponad 80% testowanych przypadków znormalizowanej krzywej Béziara generowała błąd aproksymacji poniżej 0.1. Proponowaną metodę w porównaniu z interpolacją funkcjami sklejanymi (przy tych samej liczbie użytych wielomianów), generowała błąd aproksymacji wielokrotnie mniejszy (od dwóch do nawet dwudziestu razy mniejszy).

W testach pamięciowych proponowanej metody zapotrzebowanie na pamięć było około dwa razy większe w porównaniu do zapisu krzywej Béziara trzeciego stopnia, a było w najgorszym przypadku dwa razy lepsze niż w bardzo popularnej metodzie próbkowania.

W testach wydajnościowych proponowana metoda uzyskiwała dużo lepsze wyniki niż metody obecnie stosowane (dwuetapowy algorytm do wyznaczenia interpolowanej wartości dla krzywej Béziara lub jej aproksymacji). Gdy częstotliwość próbkowania zgodna jest z częstotliwością odtwarzania, prezentowana metoda jest tylko około 2 razy gorsza niż metoda próbkowania, natomiast gdy częstotliwość odtwarzania nie pokrywa się z częstotliwością próbkowania, proponowana metoda jest już lepsza niż metoda próbkowania.

Do najważniejszych osiągnięć autora można zaliczyć:

- Wyznaczenie krzywej Béziara trzeciego stopnia w parametryzacji umożliwiającej aproksymację w ustalonej metryce funkcji ciągłych. Dla problemu opisu ruchu w animacji poszukiwana parametryzacja opisana jest w postaci funkcji zależnej od czasu. Zaproponowane rozwiązanie składa się z sześciu rodzin funkcji, a o ich końcowej postaci decydują wartości współczynników funkcji wewnętrznej g .
- Dobór odpowiedniej metryki dla rozpatrywanych warunków zadania. W metodzie przedstawionej w pracy została wykorzystana odpowiednia norma przestrzeni Sobolowa, która jest górnym oszacowaniem normy supremum. Norma ta implikuje zarówno zbieżność jednostajną jak i zbieżność punktową.
- Wyznaczenie jednoznacznego przybliżenia funkcji wejściowej przy użyciu wielomianów trzeciego stopnia, za pomocą minimalizacji funkcji opartej na normie w przestrzeni Sobolowa $W^{1,2}$. Pozwoliło to wyznaczyć jednoznacznie szukane współczynniki, bez konieczności konstrukcji odpowiedniej liczby warunków dla węzłów interpolacyjnych.

Zaproponowane rozwiązanie spełnia wszystkie założenia pracy, jednocześnie przeprowadzone testy potwierdziły poprawność uzyskanych rozwiązań. Tym samym należy stwierdzić, że

cel pracy został osiągnięty. W pracy zaproponowano metodę wyznaczenia krzywej aproksymacyjnej w odpowiedniej parametryzacji z wykorzystaniem przestrzeni Sobolewa i normy supremum. W wyniku przeprowadzonych badań potwierdzono, że możliwe jest znalezienie funkcji sklepanej trzeciego stopnia, opisanej przez niewiele kawałków wielomianowych, przybliżającej wspomnianą wyżej funkcję przestępną z dobrą dokładnością w ustalonej metryce i da się to zrobić metodami algebry.

Problem efektywnego opisu trajektorii ruchu w animacji jest nadal problemem otwartym. Wyznaczenie granicznych wartości błędów aproksymacji, dla których wielomiany aproksymujące będą akceptowalne dla obserwatora, okazało się najtrudniejszym problemem. Jednak wykracza on poza zakres zagadnień analizowanych w pracy i wymaga niezależnych badań. Jest on jednym z tych problemów, które stanowią naturalną kontynuację badań opisanych w niniejszej pracy. Z drugiej strony mimo, rozwoju sprzętu informatycznego nadal realizacja animacji w czasie rzeczywistym, z zachowaniem odpowiedniej jakości percepcyjnej, jest zadaniem bardzo trudnym. Brak jest szerokich badań nad percepcją zjawisk dynamicznych, które pozwoliłyby wypracować proste techniki oceny proponowanych rozwiązań. Zdaniem autora tematyka podjęta w niniejszej pracy jest warta kontynuacji i powinna być rozwijana.

Dodatek A

Historia animacji

A.1 Historia animacji klasycznej

Animacja jest jednym z najważniejszych elementów kultury popularnej na świecie. Widzimy ją niemal na każdym kroku. Pojawia się już nie tylko w kinach i w telewizji, ale również w internecie, reklamach, na elektronicznych wyświetlaczach rozmieszczonych w naszych miastach, a nawet na ekranach urządzeń, z których korzystamy na co dzień. Zaledwie kilka lat po upowszechnieniu telefonów komórkowych na niektórych ekranach, wtedy bardzo prostych modeli, pojawiały się animacje powitalne i pożegnalne. Animacja otacza nas z każdej niemal strony i stała się tak powszechna, że często się nad nią już nie zastanawiamy. Nie zawsze jednak towarzyszyła nam tak, jak to ma miejsce współcześnie.

W naturze ludzkiej od wieków pozostaje chęć ożywienia tego co nieożywione, kreowania i tworzenia połączonego z fascynacją otaczającym światem, jego zmiennością i ruchliwą naturą.

Już w grotach, tysiące lat przed naszą erą, nasi przodkowie malowali zwierzęta próbując odtworzyć ich aktywność. Na ścianach jaskiń powstawały serie rysunków mające odzwierciedlać kolejne fazy polowania, co, przez niektórych badaczy, porównywane jest do techniki używanej w kinematografii - do wyświetlania kolejnych klatek filmu. Trudno jest jednak uznać te prehistoryczne malowidła za zalążek animacji. Dobrze pokazują one jednak odwieczną, leżącą głęboko w naturze człowieka, fascynację ruchem i chęć jego odwzorowania w artystycznej formie.

Pierwsze urządzenia optyczne, które pozwalały uzyskać iluzję ruchu, powstały w XV wieku. Od tego czasu z mniejszym lub większym powodzeniem uczeni tworzyli lub ulepszali kolejne wynalazki często bazujące na przyrządach optycznych takich jak camera obscura. W połowie XVII wieku jezuita Athanasius Kircher zbudował i opisał w „Ars Magna Lucis et Umbrae” urządzenie, które nazwał laterna magica (z łac. latarnia magiczna). Był to bardzo prosty projektor rzutujący obraz ze szklanych, ręcznie malowanych przezroczycy. O ile pierwotnie urządzenie to miało, według założeń autora, demistyfikować zabobony i pomóc w walce z iluzjami, o tyle szybko stało się zabawką służącą właśnie do przeprowadzania różnorodnych seansów spi-

rytystycznych. Zwłaszcza w późniejszych czasach - w XVIII i XIX wieku popularne stały się pokazy podczas których zabawiano się w wywoływanie duchów. Seanse te, zwane „Fantasmagoriami”, przesycane były nastrojem grozy, a do budowania ekscytującej atmosfery używano czasem nawet kilku latarni magicznych oraz soczewek, kukiełek i różnych efektów optycznych i dźwiękowych.

W latach 30 XIX wieku powstały pierwsze urządzenia optyczne wykorzystujące właściwości percepcji dla stworzenia iluzji ruchu. Takim urządzeniem był Phenakistiscope (Fenakistiskop) – urządzenie składające się z drążka i umieszczonej na nim okrągłej tarczy, z rysunkami, przedstawiającymi kolejne etapy ruchu. Obrazki przedzielone były niewielkimi otworami, a odbiorca musiał wprawić tarczę w ruch stojąc przed lustrem i trzymając zadrukowaną stronę w kierunku jego tafli. Kiedy widz patrzył przez otwory w tarczy zmieniające się obrazki stwarzały złudzenie optyczne zlewając się ze sobą i tworząc pozory płynnego ruchu postaci.

W kolejnych latach do tego urządzenia wprowadzano udoskonalenia – w 1834 William George Horner uniezależnił urządzenie od lustra, tworząc tzw. zoetrop. To urządzenie wykorzystywało już efekt stroboskopowy. Co ciekawe efekt ten został odkryty i opisany zaledwie dwa lata wcześniej przez belgijskiego fizyka Josepha Plateau - autora fenakistiskopu.

Kilka lat później, w 1853 Franz von Uchatius umożliwił oglądanie obrazów z fenakistiskopu większej liczbie osób, dzięki wyświetlaniu ich na ekranie poprzez połączenie fenakistiskopu z latarnią magiczną.

Wszystkie te urządzenia bazowały jednak jedynie na iluzjach optycznych i właściwościach ludzkiego oka oraz percepcji.

W 1877 dzięki francuskiemu wynalazcy Émilowi Reynaud powstało pierwsze urządzenie do pokazów kinematograficznych. Powszechnie uważa się je za zwiastun współczesnej techniki filmowej.

Praksinoskop (lub praksynoskop) – to urządzenie pierwotnie składające się z lustra i rysunków umieszczonych na obrotowej, okrągłej, wymiennej podstawie. Urządzenie było niewielkie, umożliwiało oglądanie jedynie krótkich, powtarzających się sekwencji obrazków, na przykład galopującego konia. Na początku sprzedawane było jako zabawka, wkrótce jednak wynalazca udoskonalił je i powiększył o system soczewek i obrotowy bęben, w którym umieścił perforowaną taśmę wypełnioną ręcznie malowanymi obrazkami. Wszystko to umożliwiło wyświetlenie ruchomego obrazu na odpowiednio przygotowanym ekranie. Z czasem autor udoskonalał swój wynalazek, między innymi powiększając bęben, na którym umieszczał coraz bardziej rozbudowane i coraz dłuższe historie, które sam własnoręcznie, rysował. W ten sposób Emil Reynaud zapisał się w historii jako nie tylko wynalazca, ale również pierwszy animator. W 1889 roku opatentował swój wynalazek i nadał mu nazwę: „teatr optyczny”. Już 3 lata później podpisał pięcioletni kontrakt na projekcje swoich filmów w Musee Grevin - muzeum osobliwości i figurwoskowych. Pierwsze pokazy obejmowały trzy filmy, składające się z od 300 do 700 obrazków,

które w sumie tworzyły około 30 minutowy seans. Bilety na te przedstawienia nie były drogie a seanse reklamowano jako pantomimy świetlne, iluzję rzeczywistości, cud optyki i automatyki i cieszyły się one dużym zainteresowaniem publiczności.

Podczas seansów wykonywano na żywo muzykę i dialogi, co również uatrakcyjniało pokazy. Wkrótce jednak okazało się, że taśmy z rysunkami szybko się niszczą, a publiczność żądna jest nowych historii. Produkcja nowych obrazków była jednak bardzo czasochłonna, i to pomimo zastosowania metody daleko wyprzedzającej swoje czasy - wyświetlania statycznego tła i malowania jedynie elementów ruchomych w akcji.

Zanim Reynaud powrócił z nowymi historiami, publiczność zdążyła już pokochać kolejny wynalazek, częściowo bazujący na pomysłach samego wynalazcy, jednakże zaprezentowany przez Thomasa Edisona - kinetoskop. Było to urządzenie nieco przypominające fotoplastikon. Jak można przeczytać w Encyklopedii PWN:

Miał postać skrzyni z okularom i służył do indywidualnego oglądania na malutkim ekranie krótkich scenek filmowych, zarejestrowanych na przesuwanej ruchem ciągłym taśmie film. sklezionej końcami (jeden cykl projekcyjny trwał ok. 17 s), oświetlanej w stałych odstępach czasu krótkimi impulsami świetlnymi — dawało to efekt „zatrzymania” na chwilę taśmy; powstawało dzięki temu złudzenie ruchu, było jednak nie tak doskonałe, jak w późniejszym kinematografie; kinetoskop stanowił 1892–95 wielką atrakcję w USA i Europie.

W tym samym czasie nad następnym przełomowym wynalazkiem pracowali bracia Lumière. Kinematograf opatentowany został na początku 1895 roku i okazał się wynalazkiem przełomowym w historii ludzkości. Umożliwiał on zarówno rejestrację, jak i projekcję na ekranie, ruchomych obrazów dla dużej grupy widzów. W przeciwieństwie do wynalazku Reynaud, który umożliwiał wyświetlanie około 6-8 obrazków na sekundę, kinematograf działał z prędkością znacznie bardziej komfortową dla ludzkiego oka 16 klatek na sekundę. Podobieństwa łączące oba te wynalazki to między innymi użycie perforowanej taśmy, jednakże tzw. „mechanizm chwytakowy” użyty w kinematografie pozwalał na przesuwanie taśmy unieruchamiając każdą klatkę w chwili pojawienia się przed obiektywem. Jak dowiemy się z Encyklopedii PWN:

Rozwiązania te umożliwiały naświetlanie poprawnych technicznie kadrów jednakowego formatu w jednakowych odstępach. Podczas projekcji, gdy taśma przesuwała się — przy zasłoniętym obiektywie — o jedną klatkę, powidok kadru wyświetlonego łączył się z rzeczywistym obrazem następnego, jego powidok — z rzeczywistym obrazem kolejnego itd., co prowadziło do powstania złudzenia obrazu ruchomego. Tę samą zasadę wykorzystuje się we współczesnych kamerach i projektorach filmowych. Po wprowadzeniu dźwięku uległa zwiększeniu prędkość przesuwu taśmy — do 24 klatek/s.

Dzieło braci Lumière było na tyle przełomowe, że data pierwszej publicznej projekcji 28 grudnia 1895 uznawana jest za datę narodzin kina.

W ciągu zaledwie 6 lat od opatentowania przez Émila Reynauda teatru optycznego powstał wynalazek rewolucjonizujący rozrywkę. W zaledwie kilka lat pojawiały się i błyskawicznie odchodziły w niepamięć kolejne innowacje związane z optyką i rozrywką. Wiedzę o nich czerpać możemy z zarejestrowanych patentów i nielicznych resztek, które się zachowały. Sam Reynaud w przypływie rozżalenia zniszczył swój wynalazek i wrzucił wraz ze szpulami do Sekwany. Na szczęście jego syn zdołał ukryć dwa filmy i być może dzięki niemu dziś wciąż pamiętamy o historii ręcznie malowanych opowieści Émila Reynaud.

Opisując historię animacji należy mieć na uwadze, że, tak jak w wielu innych dziedzinach sztuki, nasze informacje opierają się na dokumentach i przedmiotach, które przetrwały. Jak wiele wczesnych wynalazków, zabawek optycznych, taśm i szpuli, a nawet filmów nie zniosło próby czasu, można sobie jedynie wyobrazić. W wielu muzeach i archiwach na całym świecie znaleźć można zniszczone filmy, których nie da się już odtworzyć. Taśma filmowa jest bardzo nietrwałym materiałem, a przez wiele lat nie wykonywano i nie przechowywano zapasowych kopii filmów. Nasza obecna wiedza opiera się więc głównie na tym, co zachowało się pomimo upływu czasu.

Z dokumentów historycznych możemy dowiedzieć się, że w 1900 roku Leon Gaumont (który dwa lata wcześniej założył we Francji wytwórnię filmową Gaumont, istniejącą do dziś) uzyskał francuski patent na animację poklatkową. Jednocześnie możemy znaleźć informacje o tym, że Albert Edward Smith - współzałożyciel wytwórni Vitagraph twierdził, że to on, już w roku 1898, dokonywał pierwszych udanych prób tej techniki, niestety niedocenionych przez jego współpracownika Stuarta Blacktona.

Za pierwszy film animowany wielu historyków uznaje „Apel zapałek” (ang. „*Matches Appeal*”) Anglika Arthura Melbourne-Coopera z 1899 roku. Fragment filmu przetrwał dzięki temu, że autor ukrył jego negatyw w pudełku po herbacie, w swojej kuchni. Wiele lat później, podczas przeprowadzki, Melbourne’a-Coopera przypomniał sobie o nim i dzięki temu w 1956 roku sporządzono kopie oryginalnego negatywu. W zachowanym filmie brakuje jednak początkowych ujęć - przedstawiających zapałki wychodzące z pudełka. Fragment ten został wycięty w 1908 roku przez samego autora i użyty w filmie „Magiczne zapałki”. To dobrze pokazuje stosunek twórców do wczesnych materiałów filmowych oraz obrazuje na czym możemy opierać swoją wiedzę dotyczącą tego okresu w całej kinematografii, nie tylko w historii animacji.

Pierwszy film animowany, choć trwał zaledwie minutę i 15 sekund i był robiony na zamówienie - w dzisiejszych czasach byłby uznany za kampanię społeczną lub reklamową. Musiał jednak zrobić na widzach niemałe wrażenie, gdyż jeszcze w tym samym roku pojawiły się dwa kolejne w tej serii. W Wikipedii znajdziemy następujący opis:

Jego akcja dzieje się w pustym pokoju z widocznymi po prawej stronie drzwiami; ze znajdującego się tam pudełka firmy Bryant & May wychodzą zapałki. Część z nich układa się w dwie postacie, a reszta w drabinę. Jedna postać przytrzymuje drabinę, druga wchodzi na nią

i umieszcza na ścianie tekst, stanowiący apel o fundowanie zapalek brytyjskim żołnierzom walczącym w II wojnie burskiej.

W tym samym roku powstały także dwie kolejne części filmu, stworzone przez tego samego autora „Animowane zapalki grają w siatkówkę” oraz „Animowane zapalki grają w krykieta”. Ich akcja miała miejsce w tym samym pokoju i przedstawiała tytułowe zapalki uprawiające sport.

Znanych współcześnie jest 6 filmów Arthura Melbourne’a-Coopera. Część z nich jest filmami łączącymi fragmenty grane przez aktorów z fragmentami w pełni kukiełkowymi. Ile filmów animowanych stworzył Arthur Melbourne-Cooper, nie wiadomo. Pewne jest jedynie to, że swoją pasję wraz z wybuchem I Wojny Światowej w 1914 roku porzucił na rzecz pracy inspektora w fabryce amunicji.

Do komplikacji związanych z ustalaniem chronologii powstawania filmów animowanych możemy dodać jeszcze niejasności w kryteriach analizy dokonywanych przez różnych badaczy. W zależności od kraju pochodzenia autorzy literatury specjalistycznej z zakresu kinematografii i animacji mają różne poglądy na to, co jest uznawane za film animowany. Wiąże się to najczęściej z omówionymi w poprzednim rozdziale różnicami kulturowymi w postrzeganiu pojęcia filmu animowanego, najczęściej wynikającymi z głównych nurtów panujących w danej kulturze. I tak na przykład w literaturze amerykańskiej trudno znaleźć wśród pionierów animacji Melbourne’a-Coopera - jest on często wymieniany jedynie jako specjalista w zakresie techniki poklatkowej (ang. *stop-motion*), zresztą jako jeden z wielu w tamtych czasach. Jako pierwsze filmy animowane omawiane są dopiero filmy stworzone techniką rysunkową w latach 1906-1909. Można to uznać za swego rodzaju odsunięcie animacji lalkowej na boczny tor historii filmów animowanych. Z kolei w polskiej literaturze animacja kukiełkowa nie jest tak traktowana i Paweł Sitkiewicz w książce „Małe wielkie kino. Film animowany od narodzin do końca okresu klasycznego” na kilku stronach udowadnia, że wbrew różnym niejasnościom i zarzutom niektórych badaczy, Arthur Melbourn-Cooper stworzył w 1899 roku film, który możemy nazwać pierwszym filmem animowanym.

Kolejne lata przyniosły wiele ciekawych eksperymentów w zakresie sztuki filmowej. Analizując historię kinematografii, należy brać pod uwagę nie tylko ograniczenia związane z trwałością materiałów, ale także ogromną rozpiętość geograficzną zjawisk, obejmującą wiele krajów, a nawet kontynentów. Zdarzało się, że podobne idee, eksperymenty czy innowacje pojawiały się niezależnie od siebie, w podobnym czasie, w kilku miejscach świata. Komunikacja pod koniec XIX wieku i na początku XX wieku nie przebiegała tak sprawnie jak w latach późniejszych i twórcy wielokrotnie nie wiedzieli o tym co nowego pojawia się w innej części globu. Czasem też trudno rozróżnić co było pierwotnym nurtem, a co jego kopią. Spory o to, kto pierwszy zajął się animacją kukiełek, kto stworzył pierwszy rysunkowy film animowany czy kto wynalazł technikę poklatkową, trwały latami i czasami do dziś trudno je jednoznacznie

rozstrzygnąć. Analizując historię animacji warto brać więc pod uwagę nie tylko chronologię wydarzeń, ale również ich wagę, nowatorskość i wirtuozerię opanowanej techniki.

Niewątpliwie trzech twórców wyróżniało się na początku XX wieku wśród wielu innych autorów. Pierwszym z nich był James Stuart Blackton - karykaturzysta, który w roku 1896 dostał posadę rysownika u Thomasa Edisona. Rok później wspólnie z Albertem E. Smithem założył towarzystwo filmowe Vitagraph, które w ciągu kilku lat urosło do największej wytwórni filmów niemych w Stanach Zjednoczonych. W 1906 roku Blackton zaprezentował widowni film rysunkowy pt. „Humorous Phases of Funny Faces”. Film ten opierał się na popularnych w tamtych czasach lightning sketches - filmach, w których artysta wykonuje szybkie szkice postaci. W niektórych scenach pojawiała się ręka artysty rysująca lub usuwająca z tablicy elementy postaci, w pozostałych zaś techniką poklatkową uzyskano efekt „ożywienia” twarzy rysowanych bohaterów, którzy śmieją się i dokazują. Blackton jest autorem zaledwie 4 filmów animowanych, gdyż swój czas poświęcił głównie rozwojowi wytwórni Vitagraph, jednak dzięki swoim nowatorskim dokonaniom zaliczany jest do pionierów i „ojców animacji”.

Drugim twórcą, który wpisał się do historii animacji, był Francuz - Émile Cohl, który w 1908 roku zachwyił widzów filmem „Fantasmagorie”. W tej animacji, podobnie jak u Blacktona, pojawia się w niektórych momentach ręka rysownika, ale większość tego niemal dwuminutowego dzieła opiera się na rysunkowej opowieści o szalonych, pełnych fantazji i przemian, przygodach narysowanej postaci.

Trzecim ważnym artystą wyróżniającym się w czasach pionierskich animacji był amerykańczyk Winsor McCay, który wpierw zdobył sławę jako doskonały twórca komiksów. Wniósł on do filmów rysunkowych pierwsze rozpoznawalne postacie, o określonych osobowościach, odpowiednich proporcjach i charakterystycznych ruchach. Już od pierwszego filmu „Little Nemo” z 1911 roku jego bohaterowie przestają być jedynie ożywionymi kreskami. Co ciekawe, w tym filmie pojawia się po raz pierwszy humorystyczny akcent autokreacji bohaterów - narysowany Nemo zrywa z kreacjonizmem ludzkim i sam dorysowuje kolejną postać.

Winsor McCay stworzył jeszcze kilka filmów, a każdy z nich był nowatorski i doceniany przez publiczność. Wszystkie jednak były bardzo pracochłonne i powstawały przy niewielkim udziale jedynie jednego lub dwóch asystentów. Technika stosowana przez McCaya dawała ciekawe efekty, ale nie przystawała do błyskawicznie rozwijającego się przemysłu filmowego.

W latach 1913-1915 roku kilku amerykańskich animatorów - m.in. John Bray, Earl Hurd oraz Raoul Barré - wpadło niezależnie od siebie na podobne pomysły, aby usprawnić pracę nad filmami animowanymi i zorganizować pracę studia zgodnie z nowoczesnymi zasadami produkcji przemysłowej. John Bray zaczął stosować nakładane klisze z elementami tła, Earl Hurd wykonywał kolejne rysunki na osobnych arkuszach i fotografował je na ilustracji tła, Raoul Barré natomiast wymyślił system ułatwiający animację poprzez nakładanie kolejnych warstw arkuszy z otworami.

Wszystkie te innowacje spowodowały znaczne przyspieszenie procesu powstawania filmów animowanych. Studia zaczęły zatrudniać rzesze rysowników i produkować po kilka krótkich filmów w miesiącu. Widownia pokochała filmy animowane. Kolejnym krokiem milowym w tych warunkach musiało być stworzenie pierwszego animowanego filmu pełnometrażowego.

W 1917 roku powstał pierwszy animowany film fabularny w historii. Stało się to jednak nie w kolebce nowych technologii i masowej produkcji - Stanach Zjednoczonych - a w Ameryce Południowej. Stworzył go argentyński rysownik pochodzenia włoskiego, Quirino Cristiani. Film nosił tytuł „El Apóstol”, trwał około 60 minut i był satyrą polityczną. Niestety nie zachowała się żadna jego kopia.

14 lat później Quirino Cristiani dokonał kolejnego przełomu, tworząc pierwszy film animowany ze ścieżką dźwiękową. „Peludópolis”, który trwał około 80 minut i również był satyrą polityczną odnoszącą się do ówczesnej władzy w Argentynie

Jak widać w historii animacji zapisali się nie tylko autorzy amerykańscy choć to właśnie w Stanach Zjednoczonych nastąpił niezwykle szybki rozwój techniki kinematografii. Wiele przełomowych wydarzeń jest zasługą twórców ze Starego Kontynentu, ale także z Ameryki Południowej i Azji.

W czasie, gdy w Stanach Zjednoczonych wielkie studia walczyły ze sobą o zyski, nakładając patenty na swoje wynalazki, rozbudowując i przyspieszając produkcję i tracąc jednocześnie możliwość tworzenia wartościowych, nowatorskich prac, w innych częściach świata wciąż panowała nieskrępowana wolność twórcza artystów. W czasie gdy w Ameryce tworzone szybkie, proste kreskówki ku uciechu publiki, w innych częściach świata wciąż eksperymentowano z formą. Dzięki temu powstały inspirowane chińskim teatrem cieni filmy niemieckiej animatorki Lotte Reiniger, seria przygód Kapitana Grogga autorstwa szwedzkiego rysownika Victora Bergdahla czy eksperymentalne filmy dadaisty szwedzkiego pochodzenia Vikinga Eggelunga.

Nie należy jednocześnie zapominać, że na losy nowej sztuki, tak zresztą jak na wszystkie inne, znaczący wpływ miała również historia. Tocząca się w Europie I Wojna Światowa w latach 1914-1918 niewątpliwie zahamowała rozwój wielu dziedzin, w tym animacji. Środki inwestowane były w tym czasie w wielką mobilizację wojskową, a wielu artystów, tak jak Arthur Melbourn-Cooper, porzuciło sztukę, by służyć swej ojczyźnie w bardziej praktycznych w czasie wojny dziedzinach.

Reperkusje wydarzeń historycznych szczególnie mocno odczuwalne były w Europie Wschodniej i Rosji. Mimo to jednym z najznamienitszych twórców-pionierów filmu animowanego był rosyjski filmowiec narodowości polskiej Władysław Starewicz, który już w 1910 roku we własnej domowej pracowni eksperymentował z animacją. Otaczająca przyroda inspirowała go do stworzenia pierwszego dzieła p.t. „Walka żuków jelonków”, które było odwzorowaniem zachowania żuków całkowicie stworzonym na potrzeby filmu metodą poklatkową. Jak kilka lat później opowiadał sam autor:

„Kiedy próbowałem sfilmować walkę żywych żuków jelonków o samicę, okazało się, że po zapaleniu reflektorów zamierały one w bezruchu. Wpadłem, więc na pomysł, aby uspić moich rycerzy. Oddzieliłem ich kończyny i rogi od tułowia, potem z powrotem umieściłem je na właściwym miejscu przy pomocy cieniutkich drucików. Tak spreparowane lalki z uspiionych żuków ubrałem w kostiumy, buty z cholewami, dałem do ręki rapiery.”

Dwa lata później Starewicz przyjął propozycje wytwórni Chanżonkowa i wkrótce w Moskwie odbyła się premiera filmu „Piękna Lukanida”. Film wykonany był podobną metodą do wcześniejszego dzieła, ale tym razem opowiadał mitologiczną historię Heleny i Parysa w postaci walki żuków przebranych za rycerzy. Wywołał olbrzymią sensację zarówno wśród widzów, jak i krytyków. Podobno niektórzy myśleli, że żuki były tresowane na potrzeby tego dzieła.

Następnym jego filmem lalkowym była pierwsza w historii animowana adaptacja utworu literackiego p.t. „Konik polny i mrówka” Jean de La Fontaine’a. Premiera filmu miała miejsce 15 grudnia 1912 w Kopenhadze, a jego sukces przeszedł wszelkie oczekiwania. Wykonano 140 kopii, co na ówczesne czasy było rekordem, dzięki czemu obraz obejrzeliciliczni widzowie zarówno w Europie, jak i w Ameryce.

Zimą 1914 Starewicz zwolnił się z wytwórni Chanżonkowa, aby nie utracić swojej niezależności twórczej a w roku 1919 na zawsze opuścił niespokojną w tym czasie Rosję (targaną rewolucją) i osiadł w Paryżu. Po kolejnym ogólnoswiatowym sukcesie, jaki przyszedł wraz z premierą w 1923 roku filmu „Słowiczy głos”, autor otrzymał złoty medal im. Hugona Rizenfelda za najlepszy i najbardziej nowatorski, nieamerykański film animowany wyświetlany w USA. Pomimo ofert pracy nadchodzących zza oceanu, Starewicz wolał drogę artysty niezależnego i pozostał w Paryżu, kręcąc kolejne filmy aż do śmierci w 1965 roku.

Przez wielu historyków Władysław Starewicz uważany jest za ojca animacji lalkowej, choć jak sam przyznał, już w swoich pierwszych dziełach inspirował się osiągnięciami Arthura Melbourne-Coopera. Niezaprzeczalnie jednak był jednym z pierwszych, ale co ważniejsze, jednym z najciekawszych i najbardziej kreatywnych twórców animacji lalkowej. Przetarł szlaki, inspirował kolejne pokolenia i wyznaczył trendy w kinematografii na wiele lat, a zasług jego na tym polu nie sposób pominąć.

W Europie we wczesnych latach XX wieku powstawały eksperymentalne niezależne filmy animowane. Choć w drugiej dekadzie wielu twórców mogło już inspirować się osiągnięciami innych, to jednak większość z nich, nawet powielając pomysły prekursorów animacji, modyfikowała je szukając własnej estetyki i indywidualnych środków wyrazu.

W tym samym czasie wynalazki kinematografii zawędrowały nawet w tak odległe zakątki świata jak wschodnia Azja. W 2005 roku w kolekcji filmów i projektorów w Kioto odkryto prawdopodobnie jeden z pierwszych filmów rysunkowych stworzonych w Japonii. Jego twórca pozostaje nieznany, dokładna data powstania również, ale specjaliści oszacowali, że powstał on prawdopodobnie przed rokiem 1912, czyli w czasach gdy amerykańskie filmy rysunkowe nie

były jeszcze upowszechnione w Kraju Kwitnącej Wiśni.

Pierwszym udokumentowanym pokazem zagranicznych animacji w Japonii był pokaz filmu Emila Cohla w Tokio w dniu 15 kwietnia 1912 r. W kolejnych latach twórcy japońscy, jak Ōten Shimokawa czy Seitarō Kitayama, pozostawali pod mocnym wpływem zachodnich filmów animowanych. I choć od 1917 roku w Japonii regularnie zaczęły pojawiać się krótkie filmy animowane wymienionych autorów, to inspiracji formą zachodnią nie sposób w nich nie zauważyć. Przez kolejnych kilkanaście lat japońscy twórcy, m.in. Noburō Ōfuji, Yasuji Murata, Kenzō Masaoka i Mitsuyo Seo, zgrabnie łączyli wschodnią stylistykę, tradycję i kulturę z zachodnimi trendami animacji.

Dominacja amerykańskich wzorców na polu filmów animowanych nie byłaby możliwa, bez postaci takich jak Kot Feliks, Betty Boop, Myszka Miki czy Królik Bugs. Wszystkie wymienione „osobistości” są oczywiście fikcyjne, stworzone przez rysowników, jednak ich wyraziste charaktery i szalone przygody przedstawiane w dziesiątkach kolejnych filmów sprawiły, że wymyślone, kreskówkowe osobowości podbiły serca nie tylko Amerykanów, ale również widzów na całym świecie.

Pierwsza z nich - Kot Feliks to czarno-biała, podobna do kota, postać pojawiająca się początkowo w niemych, a później również udźwiękowionych filmach rysunkowych. Został on stworzony przez amerykańskiego reżysera filmowego Pata Sullivana oraz animatora Otto Messmera i pojawił się po raz pierwszy w kreskówce pt. „Feline Follies” w listopadzie 1919 roku. Pomimo że pod koniec filmu popełnia samobójstwo, wdychając gaz, to i tak do roku 1936 powstało kilkaset kreskówek z jego udziałem, a także filmy i seriale animowane.

O zasadach rządzących rynkiem w Stanach Zjednoczonych wiele mówi historia związana z tym bohaterem. W 1922 roku, gdy Kot Feliks był już rozpoznawalny na całym świecie, wytwórnia Paramount Pictures, będąca w posiadaniu praw do tej postaci, próbowała odsunąć na boczny tor Messmera i Sullivana. Wytwórnia miała materiały, rysowników, własnych scenarzystów oraz reżyserów i mogła dalej samodzielnie produkować kolejne filmy z tej serii. Gdyby nie determinacja Pata Sullivana, straciłby on dzieło swojego życia.

Podobna historia wkrótce spotkała też Walta Disneya, który stracił prawa do stworzonego przez siebie Królika Oswalda na rzecz Studia Universal. Disney nie walczył jak Sullivan, zamiast tego stworzył kolejną niezwykłą postać: Myszkę Miki. Dopiero niemal 80 lat później, w 2006 roku, The Walt Disney Company zdołało ponownie zdobyć własność intelektualną postaci Oswalda.

Kot Felix był pierwszą postacią z kreskówek, która stała się licencjonowana. Wkrótce ten trend opanował nie tylko filmy animowane, ale i całą kinematografię w Stanach Zjednoczonych, a po jakimś czasie również w innych zakątkach świata. Twórcy w produkcjach komercyjnych zeszli na drugi plan, na pierwszym pozostały studia i ich licencje, ponieważ ten kto ma prawa do własności intelektualnej, ma z nich korzyści majątkowe. Taki kierunek rozwoju niewątpli-

wie był efektem mechanizacji procesu produkcji, w której każdy film animowany tworzą duże grupy specjalistów - kilku lub kilkunastu scenarzystów i dziesiątki a nawet setki rysowników odpowiedzialnych za niewielki i możliwie zunifikowany jedynie fragment całego dzieła.

Z dominacją wielkich firm produkcyjnych i zasadami rządzącymi na rynku, zmieniającymi twórczość w biznes, można próbować walczyć lub się do nich dostosować. Drugą drogę wybrał Walt Disney i w efekcie stworzył największą na świecie wytwórnię filmów animowanych.

Na temat życia i twórczości Walta Disneya powstało wiele prac, ma on swoich wielbicieli i krytyków, bez względu jednak na ocenę jego samego oraz, istniejącego do dziś, gigantycznego studia The Walt Disney Company, nie można odmówić mu ogromnego udziału w historii animacji.

Nie był on pionierem ani w dziedzinie animacji ani nawet innowatorem zauważalnym w historii tej sztuki. Wbrew legendzie, którą sam sobie przypisywał już za życia - nie stworzył właściwie nic przełomowego - ani pierwszy dźwiękowy film animowany nie był jego udziałem, ani pierwszy pełnometrażowy, ani nawet kolorowy. Niemniej filmy Disneya miały w sobie to coś, co sprawiło, że tłumy oglądały je w latach trzydziestych, czterdziestych i wszystkich kolejnych, aż po dziś dzień. Pierwszy pełnometrażowy film Walta Disneya - ekranizacja bajki „Królewna Śnieżka i siedmiu krasnoludków” jest nie tylko klasyką gatunku i obowiązkową pozycją dla miłośników animacji. Jest też bajką, którą z przyjemnością oglądają kolejne młode pokolenia.

Disney nie bez powodu zdobył rekordową liczbę nominacji do Oscarów i rekordową liczbę 26 statuetek oraz wiele innych nagród. Nie bez powodu też jego studio i wykreowane postacie przeżyły o dziesięciolecia własnego protoplastę. Kino przyjemne dla oka, zabawne i uniwersalne przetrwało próbę czasu a postacie takie jak Myszka Miki, pies Pluto czy Kaczor Donald są znane chyba każdemu człowiekowi w każdym zakątku globu.

Stosowana technika rysunkowa i styl Disneya (zwany wręcz klasycznym) na wiele dekad zdominowały główny nurt amerykańskich filmów animowanych. W latach trzydziestych i czterdziestych XX wieku w wyścigu o popularność mierzyło się z Disneyem co najmniej 6 innych dużych wytwórni. Wszystkie jednak postawiły na podobny schemat: serie zabawnych animacji rysunkowych (jak „Looney Tunes” w Warner Bros. lub „Happy Harmonies” w Metro-Goldwyn-Mayer) i powtarzanie w nich rozpoznawalnych postaci (Bosko, Kaczor Daffy, Popeye). Nie wiele było tu miejsca na innowacje czy kreatywność. W latach czterdziestych do panteonu bohaterów kreskówek dołączyły Tom i Jerry autorstwa Williama Hanna i Josepha Barbera (wtedy jeszcze pracujących dla wytwórni Metro-Goldwyn-Mayer), Woody Woodpecker wykreowany w Universal Studios i kilka innych postaci opartych na podobnym schemacie i podobnej technice.

Szukając innych niż rysunkowa rodzajów animacji, warto zwrócić się w kierunku Europy. To tu twórcy mieli większą wolność i możliwości eksperymentowania. Nawet niezależni artyści

mieli większe szanse przebić się i zaistnieć w świadomości odbiorców niż w opanowanej przez wielkie studia filmowe Ameryce. Niestety w latach 30 i 40 XX wieku w Europie ponownie historia i polityka znacząco zaważyły na losach wszelkich innych dziedzin.

Niemiecki malarz i pionier animacji abstrakcyjnej Oskar Fischinger był wielokrotnie kuszony przez wielkie wytwórnie zza oceanu, zdecydował się jednak na emigrację z kraju dopiero pod wpływem nazistowskich rządów, które skutecznie ograniczały wolność twórczą już kilka lat przed wybuchem II Wojny Światowej. Fischinger przeniósł się do Hollywood, gdzie najpierw podpisał umowę z Paramount Pictures, a następnie przeniósł się do studia MGM, ale nigdzie nie pozostał na dłużej. W studiu Walta Disneya też wytrzymał zaledwie kilka miesięcy, ponieważ sposób realizacji filmów, koncentracja na finansowej stronie przedsięwzięć i ograniczanie artystycznej swobody twórcy we wszystkich ówczesnych dużych studiach filmowych tak mocno doskwierały ambitnemu artyście, że wolał skoncentrować się na malarstwie i niezależnej produkcji, by już nigdy nie powrócić do współpracy z wielkimi wytwórniami.

Wraz z emigracją z targanej niepokojami Europy do Stanów Zjednoczonych przybyło w latach trzydziestych i czterdziestych wielu zdolnych twórców. Wśród nich byli doceniani artyści z ciekawym dorobkiem w dziedzinach animacji mniej popularnych w Ameryce. Takimi emigrantami byli m.in. Norman McLaren - kreatywny twórca, autor metody piksilacji oraz Węgier znany pod nazwiskiem George Pal, który uciekając przed nazistami do Ameryki przybył z opracowaną nowatorską metodą animacji poklatkowej. Polegała ona na użyciu wymiennych lalek, które aby ciekawie przedstawić ich ruch i mimikę, tworzone były całymi seriami. Był to czasopiacy i pracowity rodzaj animacji, ale efekty były tak niezwykle, że seria „Puppetoons” była nominowana w latach czterdziestych do wielu nagród, w tym nagród Akademii Filmowej, a autor został doceniony honorowym Oscarem w 1944 roku.

Wraz z końcem lat czterdziestych i nastaniem pięćdziesiątych na całym świecie zachodziły znaczne przemiany. Echa wojny ucichły, kraje odbudowywały swoje gospodarki, a nowe wynalazki, takie jak telewizja, upowszechniały się i zmieniały oblicze zarówno przemysłu rozrywkowego, jak i sztuki.

Telewizja okazała się doskonałym polem dla animacji. Krótkometrażowe filmy animowane z ulubionymi bohaterami płynnie zmieniały się na potrzeby nowego medium w rysunkowe serie. Upowszechnienie i szybki rozwój telewizji pozwoliły również na dotarcie do milionów widzów innym, bardziej oryginalnym produkcjom animowanym. Otworzyła drogę animacji lalkowej, plastelinowej, stylom mieszanym. Wciąż jednak formy inne niż klasyczna animacja rysunkowa stanowiły w Stanach Zjednoczonych niewielki procent produkcji.

Rozwój ciekawych form nastąpił w Europie. W wielu krajach powstawały studia animacji w których pracowali niezwykle utalentowani artyści. Nawet w Europie Wschodniej, gdzie narzucony był wpływ radzieckiej propagandy i kinematografii mającej służyć systemowi, powstawało wiele ciekawych projektów. Mimo że studia filmowe były kontrolowane przez państwo, a styl

wszelkiej sztuki miał mieścić się w ramach socrealizmu, autorzy wbrew wszelkim przeciwnościom, a także ograniczeniom finansowym i materiałowym, tworzyli wspaniałe i ciekawe dzieła zarówno krótko- jak i pełnometrażowe.

Dobrym przykładem będą tu efekty pracy polskiego studia Se-Ma-For. Artysta plastyk Zenon Wasilewski, pionier polskiej animacji lalkowej, malarz i fotografik, pracę nad swoim pierwszym filmem lalkowym zaczął już w okresie międzywojennym, ale dopiero po wojnie udało mu się dokończyć i pokazać publiczności film „Za króla Krakusa”. Wasilewski stworzył w swoim życiu 18 filmów animowanych i dzięki jego staraniom w Łodzi powstał Oddział Kukiełkowy w ramach łódzkiej Wytwórni Filmów Fabularnych. W styczniu 1950 r. został uruchomiony w WFF Oddział Filmów Rysunkowych, który następnie opuścił łódzką wytwórnię i przeniósł się do studia w Bielsku-Białej, a w 1956 r. Oddział Filmów Kukiełkowych został przekształcony w samodzielne Studio Filmów Lalkowych.¹

W 1958 warszawski oddział Studia Filmów Rysunkowych w Bielsku-Białej został przekształcony w Studio Miniatur Filmowych, które następnie utworzyło swój oddział w Krakowie. W roku 1980 przy poznańskim oddziale Telewizji Polskiej utworzono jeszcze Telewizyjne Studio Filmów Animowanych, które zajmowało się produkcją animowanych filmów dla dzieci i młodzieży w technikach rysunkowych, malarskich, wycinankowych, animacji w materii sypkiej i innych technikach, także lalkowych. W ten sposób w Polskiej Rzeczypospolitej Ludowej funkcjonowało aż 5 wytwórni animacji.

Od początku istnienia zarówno studio w Łodzi, Studio Filmów Rysunkowych w Bielsku-Białej jak i Studio Miniatur Filmowych w Warszawie, blisko współpracowały ze środowiskami artystycznymi, jak i szkołami wyższymi. Wielu młodych absolwentów łódzkiej Szkoły Filmowej oraz Akademii Sztuk Pięknych z całej Polski miało w ten sposób szansę na debiut i rozwój kariery. W latach sześćdziesiątych rozpoczęto realizację seriali: lalkowych „Przygody Misia Colargola” i „Piesek w kratkę” oraz rysunkowych, takich jak: „Zaczarowany ołówek” i „Proszę słonia”. W latach siedemdziesiątych i osiemdziesiątych powstawały między innymi seriale lalkowe: „Miś Uszatek”, „Trzy misie”, „Mały pingwin Pik-Pok” i „Opowiadania Muminków”; a także rysunkowe m.in. „Dziwne przygody Koziołka Matołka”, „Pomysłowy Dobromir”, „Przygody kota Filemona”, „Przygody Błękitnego Rycerzyka”, „Bolek i Lolek”, „Reksio” i „Przygód kilka wróbla Ćwirka” oraz wiele innych; plastelinowe „Plastelinki” i aktorskie, np. „Przyjaciel wesołego diabła”.

Wiele z tych filmów i seriali wyświetlanych było w krajach sąsiedzkich, jak ówczesna Czechosłowacja czy kraje nadbałtyckie wcielone do ZSSR, ale trafiały także do widzów w tak odległych zakątkach świata jak Japonia.

Wśród pełnometrażowych produkcji polskich wytwórni filmów animowanych warto wymienić serię na podstawie twórczości Tove Jansson: „Szczęśliwe dni Muminków” i „Zima w dolinie Muminków”.

W latach siedemdziesiątych przedstawiciele studia Disneya chcieli nabyć prawa do stworzenia filmowej wersji Muminków, licencji jednak nie otrzymali. Jak wspomina Antoni Bańkowski - redaktor i scenarzysta studia Se-Ma-For: „Otoczenie pani Jansson wpadło na pomysł, by zrobić serial dla dzieci. Ogłoszono konkurs, w którym brały udział studia z różnych państw. W tamtym czasie Se-ma-for był znany na świecie. Wygrała nasza koncepcja lalek płaskich, zapewne dlatego, że była bliska książkowemu pierwowzorowi i oddawała literacką atmosferę opowieści o Muminkach, a na takiej wierności oryginałowi zależy każdemu autorowi.”

W 1977 roku rozpoczęto realizację serii „Opowiadania Muminków” w bardzo nietypowej technice półpłaskiej lalki animowanej na szkłe. Polskie studia animacji wyróżniały się w świecie nie tylko kreatywnością i szerokim spektrum używanych technik, ale także dbałością o szczegóły produkcji oraz wysoki artystyczny poziom.

W latach powojennych podobny rozkwit różnorodnych form animacji obserwować możemy także w krajach, takich jak Bułgaria, Estonia, Czechosłowacja, Jugosławia, Anglia, ale również Japonia.

W latach sześćdziesiątych wyprodukowano w Japonii wiele filmów i seriali animowanych. Część z nich opierała się na tradycyjnych legendach japońskich, ale wkrótce, aby trafić do odbiorców z różnych stron świata, autorzy sięgnęli też po motywy znane z europejskich baśni. Szczególną popularnością na arenie międzynarodowej cieszyły się japońskie animacje z gatunku sci-fi, takie jak „Cyborg 009”, „Latający Okręt Widmo” i serial „Astroboy”. Oprócz animacji rysunkowych w Japonii działały także studia zajmujące się animacją lalkową i wycinankową w stylu nawiązującym do tradycyjnego japońskiego teatru lalkowego.

W Ameryce animacja lalkowa wykorzystywana była często jako element filmów aktorskich. Realistyczne animacje służyły jako efekty specjalne już w filmach takich jak „King Kong” z 1933 roku. W kolejnych latach animacja lalek rozwijała się głównie w kierunku mechanizacji i coraz bardziej spektakularnych efektów wizualnych uzupełniających filmy. W ten sposób powstał kierunek zwany „animatronics”.

Kolejny przełom w historii animacji nastąpił w chwili, gdy powstały pierwsze animacje komputerowe. Krótkie filmy realizowane za pomocą komputerów pojawiały się jako animacje niezależne od lat 70. XX wieku, choć popularność komputerowego wkładu w kinematografię (szczególnie w dziedzinie efektów specjalnych) znacznie wzrosła w latach osiemdziesiątych. Już w 1982 roku na ekrany kin wszedł film Tron, który łączył w sobie grę aktorów z obrazami generowanymi komputerowo. Pierwszym filmem pełnometrażowym zrealizowanym wyłącznie techniką animacji komputerowej był film Toy Story z 1995 roku stworzony przez firmę Pixar.

Firma Pixar Animation Studios (w skrócie Pixar) powstała w 1979 roku przy studiu Lucasfilm George’a Lucasa jako Graphics Group. W 1986 roku dział został sprzedany firmie Steve’a Jobsa i zaczął niezależną karierę pod nazwą Pixar, kontynuując badania i realizację filmów krótkometrażowych głównie demonstrujących możliwości nowych technologii. Sukces na tym

polu sprawił, że studio Walta Disneya zaproponowało firmie współpracę i w 1991 roku widzowie mieli okazję podziwiać połączenie technik komputerowych z animacją rysunkową w filmie „Piękna i Bestia”. W kolejnych filmach Disney’a, takich jak „Aladyn” i „Król lew” również pojawiają się sceny i elementy wykonane komputerowo. Przełomowy dla studia Pixar był rok 1995 i film „Toy Story” wykonany w technice animacji 3d, po którym już regularnie na ekranach kin można oglądać filmy animowane wykonane komputerowo. W 2006 roku Pixar wszedł w skład korporacji The Walt Disney Company.

Trzy lata po debiucie Toy Story na ekranach kin pojawił się pełnometrażowy film animowany 3d „Mrówka Z” konkurencyjnego studia DreamWorks. W ciągu kolejnych 20 lat te dwa studia wprowadziły na ekrany kin łącznie 47 pełnometrażowych filmów animowanych wykonanych różnymi technikami, ale zawsze wspomaganych komputerowo.

Animacje komputerowe są w tej chwili powszechnie stosowane w wielu różnych dziedzinach, nie tylko w filmie, ale także w grach, prezentacjach biznesowych, wizualizacjach oraz wielu innych. Największym wyzwaniem dla twórców wciąż jednak pozostaje fotorealistyczna animacja ruchu, zjawisk przyrodniczych oraz zwierząt i postaci ludzkich. Jak zauważył Bill Fleming: „Problem z technologią polega na tym, że potrafi ona niezwykle uprościć jedne zadania, niezwykle komplikując inne...”

A.2 Historia animacji komputerowej

Za moment przełomowy w animacji przyjęło się uważać premierę filmu „Toy Story”. Był to pierwszy film wykonany w całości techniką animacji komputerowej trójwymiarowej, jednak warto podkreślić, że animacja komputerowa sięga korzeniami dużo wcześniej, nie ogranicza się jedynie do trójwymiarowej animacji (zwanej 3D) i jest mocno zakotwiczona w pozostałych działach kinematografii.

Za pioniera w tej materii można uznać Johna Whitneyego seniora, który założył Motion Graphics Inc. w 1960 roku.

Whitney od wczesnej młodości interesował się kinematografią, w latach czterdziestych z bratem Jamesem stworzył nawet serię filmów abstrakcyjnych „Five Film Exercises”. W latach 50. Whitney wykorzystał swoje techniki animacji mechanicznej do tworzenia sekwencji do programów telewizyjnych i reklam. Jednym z jego najbardziej znanych dzieł z tego okresu była animowana sekwencja tytułowa z filmu „Vertigo” Alfreda Hitchcocka z 1958 roku.

W 1960 roku założył Motion Graphics Incorporated, w którym wykorzystywał mechaniczny komputer analogowy własnego pomysłu do tworzenia sekwencji filmowych i telewizyjnych oraz reklam. W latach siedemdziesiątych Whitney porzucił swój komputer analogowy na rzecz szybszych, cyfrowych. W latach 1969-70 eksperymentował z programowaniem grafiki komputerowej w Kalifornijskim Instytucie Technicznym, a od roku 1972 wykładał grafikę kompute-

rową na Uniwersytecie Kalifornijskim w Los Angeles.

Punktem zwrotnym w historii jego pracy był pierwszy film cyfrowy, stworzony we współpracy z Larrym Cubą, „Arabesque” - ukończony w 1975 roku. Film był raczej formą eksperymentu lub projektu badawczego dotyczącego możliwości łączenia matematyki, geometrii i grafiki na komputerze; był nieco psychodeliczny i bardzo kolorowy.

Dwa lata później Larry Cuba dostał zlecenie na przygotowanie animacji Gwiazdy Śmierci do filmu „Star Wars Episode IV: A New Hope”. Wykorzystał graficzny terminal Vector General 3D połączony z komputerem PDP-11/45, aby animacje komputerowe były wyświetlane podczas sceny odprawy pod koniec filmu. Sekwencje stworzył na Uniwersytecie Illinois, wykorzystując język programowania GRASS. W animacji występują dwie główne sekwencje. Pierwsza pokazuje Gwiazdę Śmierci jako całość, podczas gdy druga pokazuje serię widoków lotu. Z dwóch minut filmu wyprodukowanego komputerowo, w filmie pojawi się około 40 sekund. A ostatnie kilka sekund animacji, pokazujące torpedę protonową trafiającą w rdzeń reaktora Gwiazdy Śmierci, zostało narysowane ręcznie i dodane pół roku później.

W latach siedemdziesiątych grafika komputerowa była w głównej mierze efektem testów, prób i eksperymentów osób zazwyczaj związanych z wyższymi uczelniami, a jednocześnie współpracujących z branżą rozrywkową. Z tej grupy pionierów wywodził się dział Graphics Group, który powstał w 1979 roku przy wytwórni filmowej Lucasfilm. George Lucas po sukcesie „Gwiezdných Wojen” postanowił stworzyć zespół specjalizujący się w komputerowym wspomaganiu grafiki. Zadaniem zespołu było stworzenie: cyfrowego, nieliniowego systemu montażu, cyfrowego systemu edycji dźwięku oraz dalszy rozwój grafiki komputerowej. Graphics Group składał się głównie z naukowców uniwersyteckich i zespół pracował nad stworzeniem prekursora programu RenderMan, nazwanego REYES (skrót od „renders everything you ever saw” - „renderowanie wszystkiego, co kiedykolwiek widziałeś”) i opracował szereg ważnych technologii dla grafiki komputerowej, w tym „systemu cząstek” (ang. „*particle effects*”) i różnych narzędzi animacji.

W tym miejscu warto także wspomnieć o filmie sci-fi „Tron”, wyprodukowanym przez Disneya z 1982 roku w reżyserii Stevena Lisbergera, będącym jednym z pierwszych filmów wykorzystujących grafiką komputerową jako integralną część filmu. Film „Tron” ma charakterystyczny styl wizualny i chociaż nie odniósł sukcesu finansowego, stał się kultowym i jest uważany za pierwszy eksperyment w animacji komputerowej. „Tron” był także nominowany do Nagrody Akademii Filmowej (znanej także jako Oscar) w dwóch kategoriach, ale nie była to kategoria efektów specjalnych. Akademia uważała, wtedy, grafikę komputerową za zbyt łatwą w porównaniu do tradycyjnej animacji.

Graphics Group został wyodrębniony jako korporacja w 1986 roku, przy wsparciu współzałożyciela Apple Inc., Steve’a Jobsa, który stał się większościowym udziałowcem nowo powstałego studia. W tym samym czasie ukończony zostaje dwuminutowy film „Luxo Jr.”, który

staje się pierwszym animowanym komputerowo filmem nominowanym do Oscara jako Najlepszy Animowany Film Krótkometrażowy. Kolejnym krokiem miał być film pełnometrażowy, ale ambitne plany studia były niestety ograniczane przez zbyt powolny rozwój technologii. Komputery w tym czasie nie były wystarczająco potężne, by tworzyć pełnometrażowe filmy nowymi technikami. Studia zdecydowało, że w międzyczasie powinni być firmą sprzętową, a ich głównym produktem stał się Pixar Image Computer - komputer przeznaczony do projektowania grafiki, sprzedawany głównie agencjom rządowym oraz społeczności naukowej i medycznej i wykorzystywany do zaawansowanych naukowych wizualizacji w branżach takich jak medycyna, geografia czy meteorologia.

Jednym z odbiorców Pixar Image Computer było studio Walta Disneya, które użyło niestandardowego oprogramowania napisanego przez Pixar w ramach projektu Computer Animation Production System (CAPS), aby przenieść pracochłonną część malowania w procesie animacji 2D na bardziej zautomatyzowaną metodę. Na początku lat dziewięćdziesiątych firma Pixar opublikowała niektóre z narzędzi oprogramowania na otwartym rynku dla systemów Macintosh i Windows. RenderMan był jednym z wiodących pakietów 3D z początku lat 90.

„Naszym celem jest uczynienie z Rendermana i Icemana oprogramowania systemowego lat 90.”

- przyznał w jednym z wywiadów Steve Jobs

Jak pisze Maciej Bortnowski w artykule: „20 Lat Toy Story - czyli o animacji komputerowej słów kilka”:

Eksperci filmowi wskazują 3 utwory, które już w połowie lat 90 sygnalizowały rewolucję cyfrową w produkcji: „Toy Story”, „Park Jurajski” z przekonującymi i realistycznymi dinozaurami i „Terminator 2” ze spektakularną animacją T-1000 za pomocą morfingu – RenderMan był użyty przy produkcji każdego z nich.

Jeszcze w 1994 r. Steve Jobs planował sprzedać Pixar innym firmom, takim jak Hallmark Cards, współzałożycielowi firmy Microsoft Paulowi Allenowi lub dyrektorowi firmy Oracle Larry'emu Ellisonowi. Dopiero po prognozach, które sugerowały, że „Toy Story” będzie prawdopodobnie hitem - i po potwierdzeniu, że Disney rozprowdzi go w sezonie świątecznym w 1995 r. - zdecydował się dać Pixarowi kolejną szansę. Prognozy krytyków w tym przypadku okazały się trafne, film stał się sensacją, zyski były ogromne a akcje firmy Pixar wystrzeliły na giełdzie. Od tamtej pory Studio Pixar wyprodukowało 20 pełnometrażowych filmów animowanych i zdobyło dziewiętnaście Oscarów, osiem Złotych Globów i jedenaście nagród Grammy oraz wiele innych nagród i wyróżnień. W 2019 roku na ekrany kin ma wejść film „Toy Story 4”.

Mimo niezwyklej popularności animacji komputerowych, zwłaszcza animacji 3d, wciąż powstają filmy tworzone tradycyjnymi metodami animacji, jak na przykład francusko-brytyjski

film „Iluzjonista” z 2010 roku. Współczesnemu widzowi czasami może być trudno odróżnić animacje tworzone komputerowo od tradycyjnych. Filmy, które wyglądają jak tradycyjne, odręczne animacje, są wspomagane komputerowo. Często te dwie techniki przenikają się i uzupełniają. Popularny serial animowany „Family Guy” jest rysowany ręcznie, lecz nie na papierze, a za pomocą tabletów graficznych.

Zdarza się również, że tradycyjne filmy animowane wspierane są komputerowo w celu uzupełnienia klatek, by uzyskać wrażenie większej płynności ruchu. Z kolei filmy animowane komputerowo stylizowane są celowo na wzór tradycyjnej animacji.

Bez względu na to, w jakiej technice film powstaje, kluczowy jest jego odbiór przez widzów. Walt Disney twierdził, że bohaterowie są najbardziej istotni w filmach - ten dogmat pozostaje niezmienny od stu lat. Animacja z jednej strony rządzi się własnymi prawami, ale są one jednak często zbieżne z zasadami panującymi w innych dziedzinach kinematografii. Zasadniczo ani technika, ani przeznaczenie animacji, nie mają znaczenia - ważne jest stworzenie produkcji, która przyciągnie widzów i zostanie w ich pamięci.

Dodatek B

Techniki i metody animacji

Różnorodność gatunków i technik wykorzystywanych w animacji znacznie utrudnia dokonanie ścisłej klasyfikacji. Jak w wielu innych sztukach różne media i elementy mogą być wykorzystywane w jednym dziele. Próbując dokonać systematyzacji można animacje dzielić ze względu na technikę wykonania lub technikę filmowania i kreowania.

Podział animacji ze względu na technikę tworzenia:

- **Rysunkowa.**

Klatki filmu są rysowane zazwyczaj na celuloidach lub na kartkach, kalkach itp., współcześnie mogą być wykonywane przy użyciu odpowiednich programów komputerowych. Możliwe jest animowanie na warstwach, poprzez układanie na szybach elementów takich jak tło, ludzie, zwierzęta.

- **Wycinankowa.**

Powstaje poprzez wprowadzanie zmian do znajdującej się w poszczególnych klatkach kompozycji odpowiadających kolejnym fazom ruchu. Przedmiotem animacji są wycięte kawałki papieru przedstawiające fazy ruchu. Podobnie jak animacja rysunkowa, może powstawać na warstwach.

- **Lalkowa (przedmiotowa).**

W której otoczenie i postacie wykonuje się jako rzeczywiste przedmioty trójwymiarowe. Animowana postać posiada szkielet odpowiadający własnościom anatomicznym i umożliwiający stabilne ustawianie kolejnych faz ruchu. Na tej konstrukcji budowana jest odpowiadająca charakterystyce postaci forma „ciała”. Najczęściej szkielet jest konstrukcją metalową, a pozostałe elementy robi się z różnych materiałów naturalnych (bawełna, drewno) lub syntetycznych (gąbka, PVC, lateks, silikon).

- **Plastelinowa.**

Polegająca na modyfikacji postaci i wykonanego z materiałów plastycznych otoczenia metodą poklatkową, jest to jedna z form techniki lalkowej.

- **Animacja materiałów sypkich.**

Przedmiotem animacji jest np. mąka, sól, cukier, piasek. Animacja polega na przesuwaniu i animowaniu takich właśnie materiałów.

- **Pikselacja, fotoanimacja.**

W tym typie animacji wykorzystuje się wcześniej sfotografowane przedmioty, zwierzęta lub postacie ludzkie w różnych fazach ruchu. Film „*Tango*” Zbigniewa Rybczyńskiego zrealizowany taką techniką w 1980 roku w Se-Ma-Forze trzy lata później zdobył Oscara dla krótkometrażowego filmu animowanego.

- **Animacja trójwymiarowa.**

Tworzona w technologii cyfrowej (komputerowej). Powstaje w programach do przetwarzania grafiki trójwymiarowej.

- **Techniki mieszane (kolaż) i specjalne.**

W tym przypadku materiały wykorzystywane do tworzenia animacji mogą być bardzo zróżnicowane: od materiałów sypkich do animacji przedmiotowej i wycinankowej. Główne założenie opiera się na łączeniu technik w celu uzyskania dodatkowych efektów, niedostępnych w przypadku korzystania tylko z jednego z wcześniej zdefiniowanych typów animacji.

- **Animacja rotoskopowa.**

W niej wykorzystuje się uprzednio przygotowane filmy nakręcone przy użyciu kamer (najczęściej jest to ruch postaci ludzkich lub obiektów organicznych), następnie obraz jest przenoszony klatka po klatce na medium, w którym powstaje odpowiadający im film animowany.

Dodatek C

Rodzaje animacji komputerowej

W animacji komputerowej wyróżnia się następujące metody wykorzystywane podczas jej produkcji:

- **Metoda klatek kluczowych** (ang. *keyframing*).

Ma korzenie w tradycyjnej animacji rysunkowej i polega na wyznaczaniu przez animatora wartości parametrów, poddawanych animacji w tak zwanych klatkach kluczowych, natomiast wartości pośrednie między klatkami kluczowymi obliczane są przez algorytm komputerowy. W tej metodzie artysta ma kluczową rolę oraz bardzo dużą kontrolę nad powstawaniem animacji.

- **Animacja proceduralna** (ang. *procedural techniques*).

Jest rodzajem animacji komputerowej służącej do automatycznego generowania animacji w czasie rzeczywistym. Wykorzystuje się do tego celu proste wzory matematyczne, które sterują animacją. Dobrymi przykładami wykorzystania tej metody są animacje oparte o kinematykę prostą lub odwrotną [70] oraz animację ruchu postaci przy użyciu kilku prostych zależności matematycznych pokazanych w [29].

- **Animacja behawioralna** (ang. *behavioral animation*).

Bierze pod uwagę indywidualne zachowanie się istot żywych podczas ruchu oraz ich interakcje z innymi żywymi istotami. Pozwala animować nie tylko pojedynczą jednostkę, lecz całą ich grupę. Przykładami tego rodzaju animacji są między innymi: modelowanie stada, animowanie zachowania drapieżnik-ofiara.[26]

- **Animacja fizyczna** (ang. *physically based animation*).

Ta metoda jest rozwinięciem metod proceduralnych. Znając matematyczne podstawy zjawisk fizycznych świata rzeczywistego, przeważnie w formie równań różniczkowych. Moc komputerów, choć nie zawsze wystarczająca, pozwala animować obiekty zgodnie z prawami fizyki. Do symulacji fizycznie prawdopodobnych ruchów obiektów wykorzystuje się tu numeryczne metody przybliżonego rozwiązywania równań różniczkowych. Dzięki

tym technikom możliwa jest animacja dynamiki np. włosów i futra, symulacja wody oraz tkanin, dynamika ciała sztywnego.[54]

- **Animacje sterowane sztuczną inteligencją** (ang. *AI driven animation*).

Metody te także stanowią rozwinięcie metod proceduralnych. Wykorzystują one sieci neuronowe, które poddane uprzednio procesowi uczenia wykorzystywane są do wspomagania kontroli ruchów postaci [11] lub generowania animacji w czasie rzeczywistym dla uzyskania realistycznej animacji symulującej zachowanie np. istoty ludzkiej. [51][52]

- **Przechwytywanie ruchu** (ang. *motion capture*).

Metoda polegająca na rejestracji ruchu obiektów (zwłaszcza ludzi) w przestrzeni trójwymiarowej. Pozwala ona na osiągnięcie wysokiego poziomu realizmu. Dane o kolejnych fazach położenia obiektu zbierane są przy wykorzystaniu różnego rodzaju znaczników, na śledzonym obiekcie, umożliwiających przestrzenną lokalizację punktów. Są one następnie wprowadzane do oprogramowania i wykorzystywane jako odniesienie dla ruchu animowanych postaci.[53]

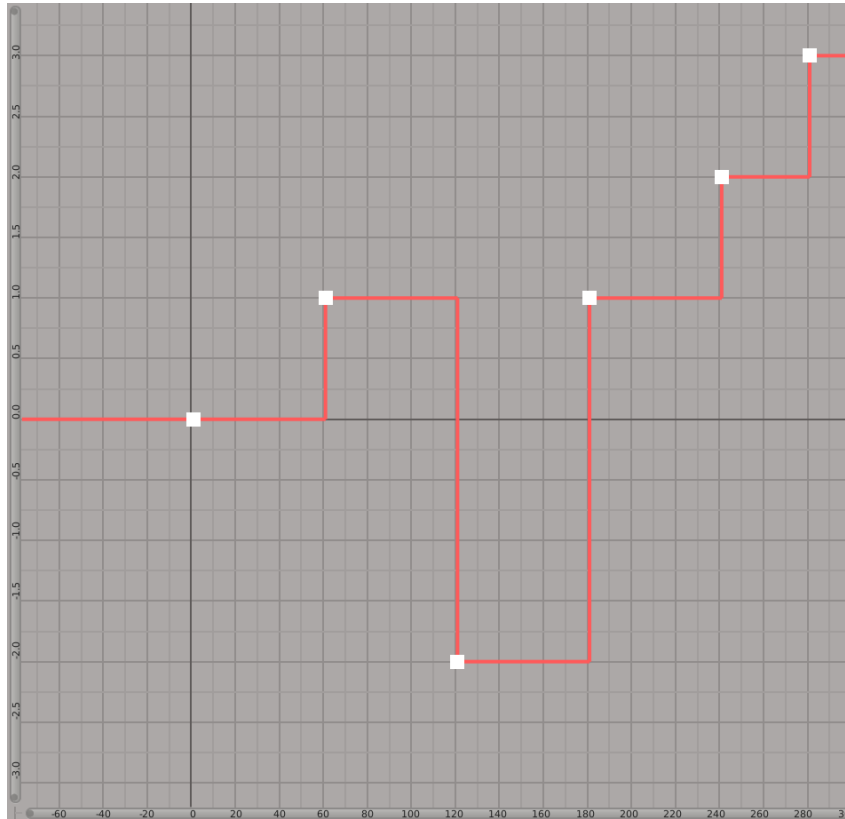
Dodatek D

Rodzaje funkcji interpolacyjnych

Programy graficzne, które zawierają moduł odpowiedzialny za budowanie animacji przy użyciu graficznego edytora, udostępniają użytkownikom różnego rodzaju funkcje interpolacyjnych. Najczęściej spotykanymi funkcjami są.

- **Funkcja schodkowa**

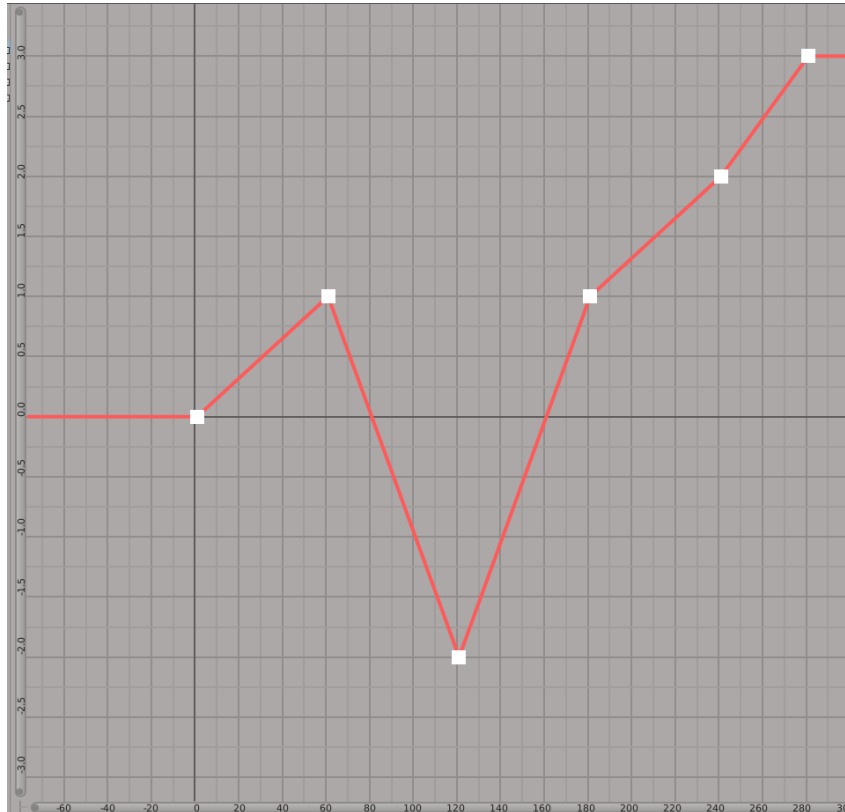
Interpolacja przy pomocy funkcji schodkowej (czyli odcinkami stałej) nie nadaje się do animowania dowolnego parametrów, gdyż nie gwarantuje zachowania ciągłości C^0 między sąsiadującymi funkcjami. Skoki (czyli różnice wartości na końcach sąsiednich przedziałów) przeważane nie są akceptowalne z punktu widzenia „płynności” animacji. Jednakże dzięki nim powstaje możliwość symulowania takich efektów jak m.in. ruch wskazówek zegara czy też teleportacja. Co ważniejsze, taki sposób interpolowania pozwala animować parametry, które nie są ciągłe w swej naturze, na przykład numer indeksu klatki w animacji „duszków” (ang. *sprite animation*).



Rysunek D.1: Wykres funkcji skokowej w edytorze graficznym Blender.

- **Funkcje kawałkami liniowe**

Interpolacja kawałkami liniowa jest jednym z podstawowych i najprostszych sposobów interpolacji. Jak sama nazwa wskazuje, wartości między kluczami są tu interpolowane funkcjami liniowymi na poszczególnych odcinkach (ze stałą prędkością pomiędzy poszczególnymi klatkami kluczowymi).



Rysunek D.2: Wykres funkcji kawałkami liniowej w edytorze graficznym Blender.

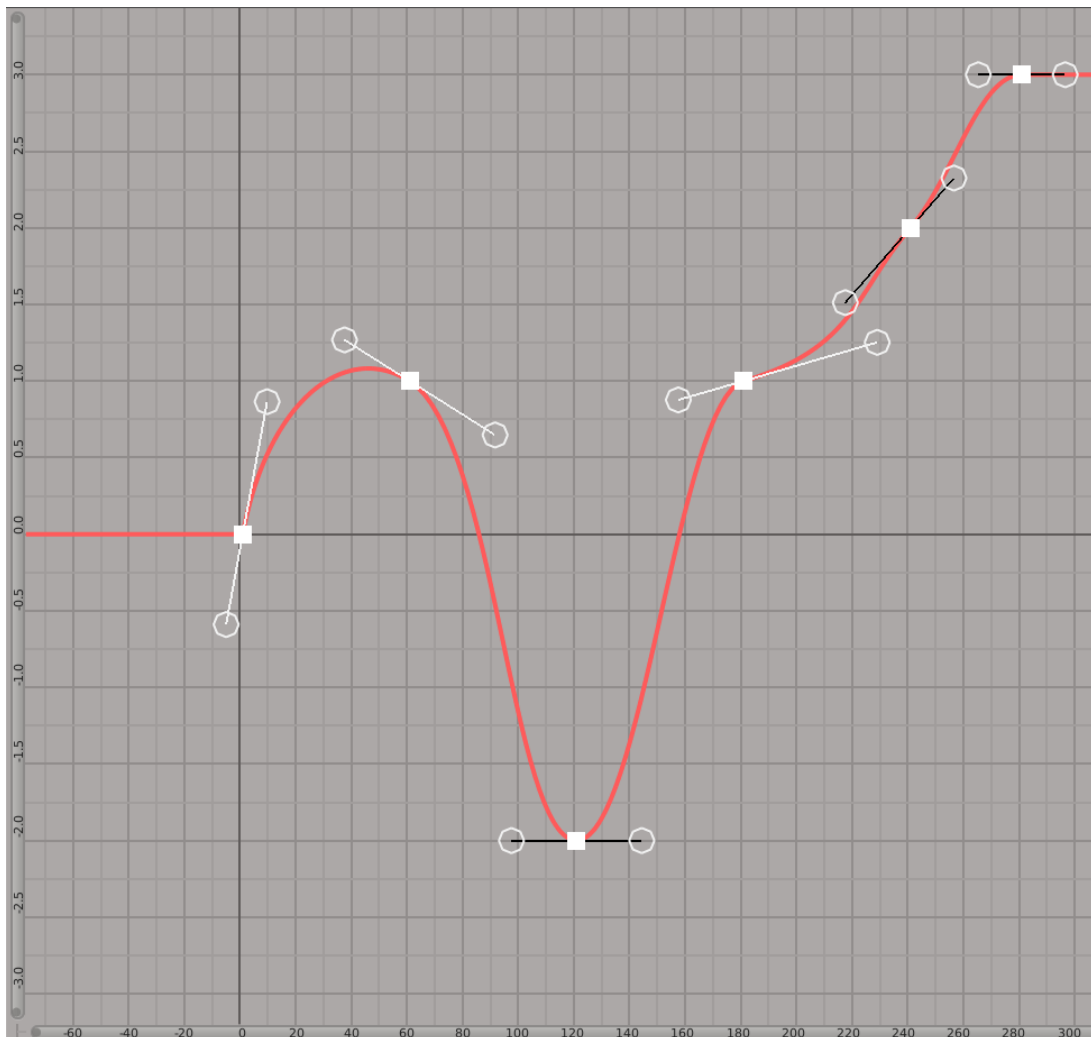
- **Krzywa Béziera trzeciego stopnia**

W latach sześćdziesiątych XX wieku w fabryce Renault rozpoczął pracę pierwszy system modelowania geometrycznego na użytek komputerowego wspomaganie projektowania. Twórcą tego systemu był Pierre Bézier. Reprezentacja krzywych, która była opracowana właśnie do tego celu, modelowania geometrycznego została nazwana krzywą Béziera. Warto jednak pamiętać, że krzywa ta niezależnie została opracowana przez dwóch inżynierów, drugim był Paul de Casteljau. Tak naprawdę pierwszy był Paul de Casteljau, ale firma Citroën, w której pracował, nie opublikowała jego opracowania. Pierre Bézier pracował w Renault i dowiedział się, że jest ciekawy pomysł de Casteljau. Natomiast popularyzację rozwiązania zawdzięczamy zupełnie komuś innemu. Zrobił to A.R. Forrest w 1972 roku [9], który pokazał, że krzywa Béziera jest to reprezentacja używająca bazy wielomianów Bernsteina.

Mimo że reprezentacja krzywej była wykorzystywana do modelowania geometrycznego [4] [8], to w edytorach graficznych bardzo często wykorzystuje się ją do interpolacji w metodzie klatek kluczowych, gdyż, w odróżnieniu od interpolacji liniowej, pozwala interpolować parametry ze zmienną prędkością, a nawet ze zmiennym przyspieszeniem.

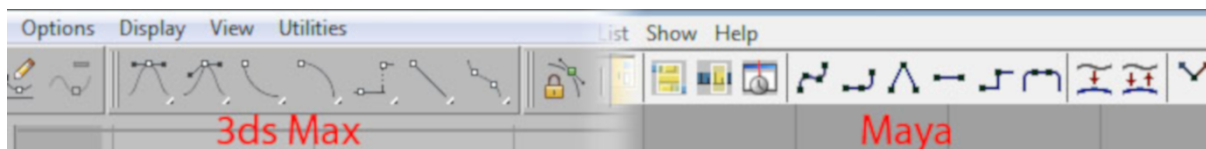
Cztery punkty kontrolne P_0, P_1, P_2, P_3 definiują krzywą Béziera trzeciego stopnia. Punkty P_0 oraz P_3 należą do krzywej i pokrywają się z klatkami kluczowymi w animacji. Dwa po-

zostałe punkty nie należą do krzywej i określone są jako uchwyty, a przemieszczanie ich powoduje zmianę wyglądu krzywej na całej jej długości, co pokazane zostało na rysunku D.3.



Rysunek D.3: wykresy krzywych Béziera trzeciego stopnia w edytorze graficznym Blender. Pomarańczowe kwadraty reprezentują punkty kontrolne pokrywające się z klatkami kluczowymi animacji, a białe okręgi reprezentują uchwyty poszczególnych krzywych Béziera.

Dodatkowo, programy graficzne mają funkcje, które ułatwiają użytkownikom kontrolę zachowania ciągłości między sąsiadującymi krzywymi Béziera poprzez powiązanie ze sobą odpowiednich punktów kontrolnych sąsiednich krzywych, udostępniając odpowiednie kontrolki w interfejsie użytkownika (rys. D.4).



Rysunek D.4: Zdjęcie części interfejsów graficznych programów 3ds Max oraz Maya, które mają za zadanie ułatwienie użytkownikowi kontroli nad ciągłością krzywej Béziera w klatkach kluczowych.

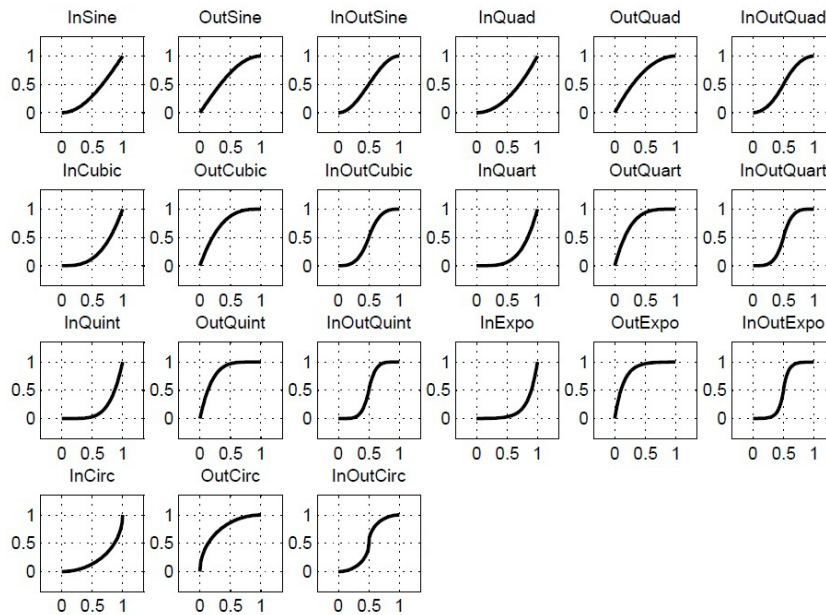
Takie powiązania uchwytów powodują, iż modyfikacja uchwytu jednej krzywej automatycznie powoduje odpowiednią zmianę odpowiedniego uchwytu drugiej krzywej tak, aby zachowana została ciągłość klasy C^1 lub C^2 .

Innym narzędziem dostępnym dla użytkownika jest ustawianie uchwytów w z góry określonej pozycji. Pozycje powodują, że krzywa Béziera przybiera kształt gładkiego startu (ang. *Ease-In*), gładkiego hamowania (ang. *Ease-Out*) czy też kształtu linii prostej.

Dokładniejszy opis krzywych Béziera i problemy z jej użyciem jako funkcji interpolacyjnej, w metodzie klatek kluczowych, przedstawiony został w rozdziale 3.1.

- **Funkcje wygładzające (funkcje Pennera [27])**

W świecie rzeczywistym procesy, takie jak ruch, rzadko mają charakter skokowy. Ruch obiektów rzeczywistych nie rozpoczyna się, ani nie kończy się nagle. Dlatego zestaw zaproponowany przez Roberta Pennera [27] zawiera 21 funkcji wygładzających (ang. *easing functions*), pokazanych na rysunku D.5, symulujących poruszanie się obiektów ze zmienną prędkością w przyjemny dla oka sposób. Funkcje te pogrupowane zostały na typy o wspólnych cechach, takich jak: gładki start (ang. *Ease-In*), gładkie hamowanie (ang. *Ease-Out*), gładki start oraz hamowanie (ang. *Ease-In-Out*). Biblioteki Qt[57] oraz Windows Presentation Foundation [58] implementują dodatkowy typ funkcji wygładzających gwałtowny start i hamowanie (ang. *Ease-Out-In*), który nie został uwzględniony w pracy Pennera.

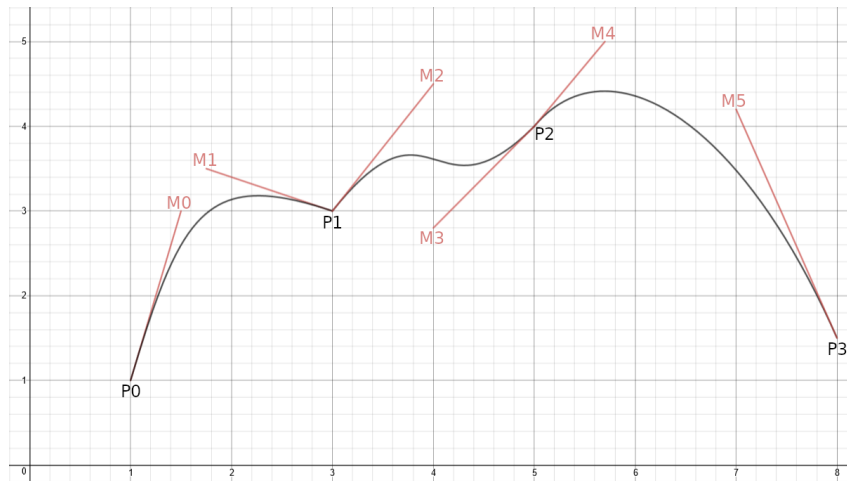


Rysunek D.5: Wykresy funkcji wygładzających, gdzie kolumna pierwsza i czwarta przedstawia funkcje typu gładki start, kolumna druga i piąta to funkcje typu gładkie hamowanie, a kolumny trzecia i szósta to funkcje typu gładki start oraz hamowanie.

Funkcje z rysunku D.5 są to funkcje wielomianowe (na rysunku z przyrostkami *Quad*, *Cubic*, *Quart*, *Quint*) lub też funkcje przestępne (z przyrostkami *Sine*, *Expo*, *Circ*). Więcej na temat opisu matematycznego przedstawionych funkcji wygładzających można znaleźć w pracy [13].

- **Krzywe Hermite’a trzeciego stopnia**

Krzywa interpolacyjna Hermite’a (gdzie wzór opisany został w (1.2)) to krzywa łącząca dwa dane punkty (tak samo jak w interpolacji Béziera), ale w odróżnieniu od interpolacji Béziera użytkownikowi udostępnione są do modyfikacji wektory styczne w punktach (co pokazane jest na rysunku D.6).



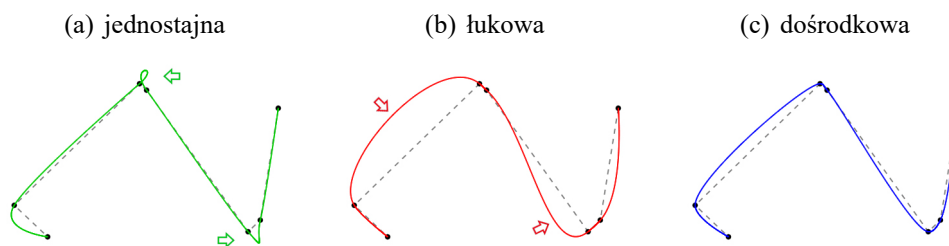
Rysunek D.6: Krzywa Hermite'a trzeciego stopnia reprezentowana przez punkty P_0, P_1, P_2, P_3 , wraz z wektorem stycznej $M_0, M_1, M_2, M_3, M_4, M_5$.

Ciągłość krzywej sklejanej, zbudowanej z połączonych krzywych Hermite'a, można zapewnić poprzez podanie wspólnych końców i wspólnych wektorów pochodnych w tych punktach. Konstrukcja takiej krzywej może być bardzo pracochłonna, zwłaszcza, gdy użyta jest większa liczba krzywych Hermite'a w animacji.

- **Krzywe Catmulla-Roma**

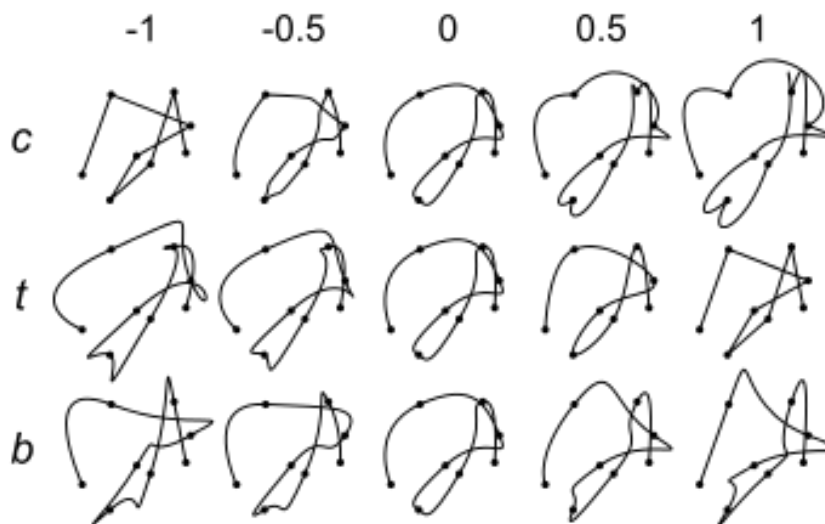
Krzywe Catmulla-Roma są szeroko wykorzystywane w grafice do różnorodnych zastosowań, od modelowania aż po animacje [5]. Krzywa Catmulla-Roma (o wzorze (1.4)), w przeciwieństwie do krzywej Béziera, interpoluje wszystkie punkty kontrolne oprócz dwóch skrajnych. Wektor stycznej w każdym punkcie interpolacji równy jest połowie wektora wyznaczonego przez wektor zbudowany z następnego punktu kontrolnego minus poprzedni punkt kontrolny. Takie wymuszenie powoduje, że łączenie krzywych zachowuje klasę gładkości C^1 w punktach interpolacji.

Konstrukcja krzywej Catmulla-Roma, zaproponowana w pracy [5], zwana parametryzacją jednostajną (ang. *uniform*), może prowadzić do powstania krzywych, które zawierają pętle oraz samoprzecięcia, przechodząc przez punkty kontrolne (jak na rysunku D.7). Dlatego też zostały opracowane inne parametryzacje krzywej Catmulla-Roma [44]. Są nimi parametryzacje: łukowa (ang. *chordal*) oraz dośrodkowa (ang. *centripetal*), służące do minimalizowania, nie zawsze pożądanego efektów, np. samoprzecięcia (co pokazane jest na rysunku D.7).



Rysunek D.7: Krzywe Catmull-Roma w trzech parametryzacjach.

- **Krzywe TCB** Krzywe TCB autorstwa D. Kochanka i R. Bartelsa [17] są uogólnieniem krzywych Catmulla-Roma. Postać krzywej wyznaczona jest na podstawie wartości trzech parametrów: napięcia (ang. *tension*), gładkości (ang. *continuity*) oraz skosu (ang. *bias*). Wpływ wartości tych zmiennych na wygląd krzywej przedstawiono na rysunku D.8.



Rysunek D.8: Przykłady wpływu wartości parametrów c-Gładkości, t-Napięcia, b-Skosu na wygląd krzywych Kochanka-Bartelsa.

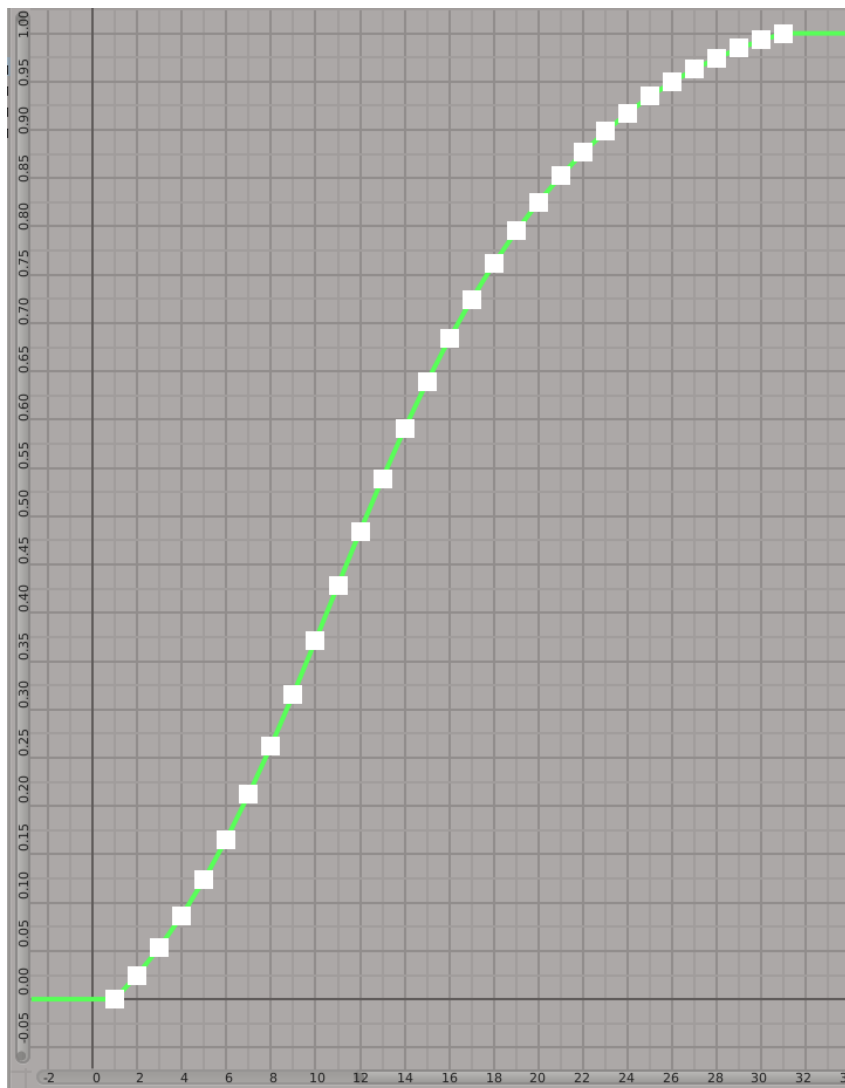
Warto zauważyć, że gdy każdy z tych trzech parametrów, przybiera wartość 0 wówczas krzywa Kochanka-Bartelsa będzie tożsama z krzywą Catmulla-Roma.

Programy graficzne do edycji animacji komputerowej metodą klatek kluczowych mają pełną dowolność w wyborze możliwych funkcji interpolacyjnych, jakie udostępniają użytkownikom. I nie wszystkie opisane wyżej funkcje są implementowane w programach graficznych takich jak Blender, 3ds Max czy Maya.

Dodatek E

Próbkowanie

Wszystkie funkcje interpolacyjne, bez względu na rodzaj, poddaje się procesowi próbkowania z zadaną częstotliwością. W silnikach gier wideo problem interpolacji krzywych często jest rozwiązywany poprzez próbkowanie, z pewną częstotliwością, wejściowej krzywej interpolacyjnej (co pokazano na rysunku E.1). Konsekwencją próbkowania krzywej interpolacyjnej jest stabilizowanie krzywej (z zadaną częstotliwością). Dane stabilizowane mogą być pomocne do wyznaczenia funkcji sklepanej pierwszego stopnia, klasy C^0 , do aproksymacji funkcji wejściowej.



Rysunek E.1: Rysunek przedstawiający ideę próbkowania krzywej interpolacyjnej, na podstawie programu Blender.

To podejście, po pierwsze, powoduje, że znika problem numerycznego wyznaczenia wartości dla danego czasu w interpolacji lub znalezienie przeciwobrazu zadanej wartości, gdzie funkcja jest wielomianem co najwyżej trzeciego stopnia. Po drugie, jest możliwość wyeliminowania interpolacji krzywej w ogóle, gdyż częstotliwość próbkowania równa się częstotliwości generowania obrazu, wartości funkcji interpolującej można pobierać bezpośrednio danych wcześniej stabilizowanych.

Gry wideo wyświetlane są na monitorach, które odświeżają wyświetlany obraz z pewną konkretną częstotliwością (obecnie dominującą wartością jest 60Hz).

To rozwiązanie, eliminując jeden problem, generuje inny, jakim jest zwiększone zapotrzebowanie pamięciowe na zapis interpolacji parametrów. Gdy liczba parametrów wykorzystana w animacji obiektów sięga wspomnianej wcześniej wartości 10^5 , to dla jednej sekundy animacji obiektów zapotrzebowanie wynosi około 23 megabajtów. Następnym problemem może

być powtórna edycja próbkowanej krzywej, gdyż próbkowanie zawężyło w znaczącym stopniu kontrolę nad krzywą z globalnej na lokalną.

Silniki gier wideo, by zmniejszać zapotrzebowanie pamięciowe na zapis animacji, często udostępniają różnego rodzaju kompresje animacji. I tak na przykładzie silnika Unreal Engine 4 [65] można wymienić następujące przykłady kompresji:

- **Kwantyzacja danych** (ang. *quantization*).
Ten proces ma za zadanie zmniejszenie liczby bajtów potrzebnych do zapisania liczb zmiennoprzecinkowych.
- **Usuwanie co drugiego klucza** (ang. *remove every second key*).
Ten proces zmniejsza częstotliwość próbkowania o połowę np. z 60Hz na 30Hz.
- **Usuwanie trywialnych kluczy** (ang. *remove trivial keys*).
Algorytm analizuje punkty kontrolne i usuwa te, które nie zmieniają wartości na danym odcinku czasu.
- **Usuwanie liniowych próbek** (ang. *remove linear keys*).
Algorytm analizuje punkty kontrolne i usuwa te, które zmieniają wartości w sposób liniowy na danym odcinku czasu.
- **Dopasowanie krzywej** (ang. *curve fitting*).
Zamiast zapisywać próbki wygenerowane przez próbkowanie krzywej interpolacyjnej, interpolacja zapisywana jest przez funkcję, której zapis wymaga mniej miejsca w pamięci niż zapis próbek.

W procesie dopasowania krzywej można zastosować znane metody interpolacji, np:

- **Interpolacja wielomianowa**

Interpolacja wielomianowa, nazywana też interpolacją Lagrange'a [62], jest to metoda, która wykorzystuje wielomian N -go stopnia w interpolacji $N + 1$ węzłów. Węzły mogą być wyznaczane z daną częstotliwością. Zgodnie z twierdzeniem Weierstrassa [40], dowolną funkcję $y = f(x)$ ciągłą na przedziale domkniętym można dowolnie przybliżyć za pomocą wielomianu odpowiednio wysokiego stopnia.

Warto jest zaznaczyć że wielomian interpolujący nie musi być (i prawie nigdy nie jest) optymalnym wielomianem aproksymującym.

Interpolacja wielomianowa w przypadku, gdy próbki są równo oddalone od siebie, może powodować pogorszenie jakości aproksymacji, mimo zwiększenia liczby jej węzłów. To zjawisko nazwane jest efektem Rungego [33]. Początkowo wraz ze wzrostem liczby węzłów N przybliżenie poprawia się, jednak po dalszym wzroście N , zaczyna się pogarszać, co jest szczególnie widoczne na końcach przedziałów. Takie zachowanie się

wielomianu interpolującego jest zjawiskiem typowym dla interpolacji za pomocą wielomianów wysokich stopni przy stałych odległościach węzłów.

Ponieważ zgodnie z twierdzeniem Weierstrassa istnieje ciąg interpolujących wielomianów coraz wyższych stopni, które przybliżają jednostajnie funkcję ciągłą, można uważać to za paradoks, iż efekt Rungego ma dokładnie odwrotny wynik. Jest to spowodowane brakiem zbieżności szeregu Taylora do funkcji aproksymowanej w przedziale aproksymacji.

• Interpolacja funkcjami sklejanymi

Powszechnie stosowaną metodą aproksymacji funkcji jest metoda interpolacji funkcjami sklejanymi, zwanymi także splajnami (ang. *spline*). Funkcja sklejana k -ego stopnia S_k to wielomiany k -ego stopnia, które posiadają tę właściwość, że interpolują wartość wejściowej funkcji w $n + 1$ punktach $x_0, x_1, \dots, x_{n-1}, x_n$ (nazywanych węzłami) oraz mają ciągłe pochodne $k - 1$ rzędu w węzłach wewnętrznych x_1, \dots, x_{n-1} .

Interpolacja funkcjami sklejanymi jest często preferowana do interpolacji wielomianowej, ponieważ błąd aproksymacji może być niewielki nawet przy użyciu funkcji sklejanych niskiego stopnia. Ponadto pozwala uniknąć efektu Rungego. Dodatkowo można wykazać [33], że zwiększanie liczby węzłów skutkuje zmniejszaniem błędu aproksymacji funkcji aproksymowanej względem normy supremum (błąd dąży do 0, dla funkcji klasy C^2 lub C^1 , której pochodne spełniają warunek Lipschitza).

Warto zaznaczyć, że funkcje sklejane pierwszego rzędu są tożsame z interpolacją funkcjami liniowymi. Najczęściej w praktyce stosowane są splajny trzeciego stopnia, co w tym przypadku powoduje, że funkcja sklejana reprezentowana jest przez wielomian trzeciego stopnia oraz pierwszą i drugą pochodną w węzłach. Zachowują gładkość klasy C^2 .

Do jednoznacznego wyznaczenia współczynników N wielomianów co najwyżej trzeciego stopnia potrzebne jest rozwiązanie $4 * N$ równań. Interpolacja wartości w węzłach wymaga $2N$ równań. Aby funkcja interpolowała wartości w węzłach, konieczne są $2N$ równań. Ponadto potrzeba $N - 1$ równań, by zapewnić gładkość klasy C^1 , oraz $N - 1$ do zachowania gładkości klasy C^2 . Brakujące 2 równania można określić dowolnie. Równania te mają nazwę warunków brzegowych i wyróżnić można dwa warunki, które generują

– naturalne funkcje sklejane (ang. *natural spline*)

występuje wtedy gdy spełniony zostanie warunek

$$\begin{cases} S_3''(x_0) = 0 \\ S_3''(x_N) = 0 \end{cases}$$

– zaciśnięta funkcja sklejana (ang. *clamped spline*)

$$\begin{cases} S'_3(x_0) = f'(x_0) \\ S'_3(x_N) = f'(x_N) \end{cases},$$

gdzie f' to pochodna funkcji aproksymowanej.

Dodatek F

Nierówność Sobolewa dla funkcji określonych na odcinku

Dodatek ten zawiera kilka elementarnych faktów z teorii przestrzeni Sobolewa. Przestrzenie te najczęściej oznaczane są symbolem $W^{n,p}(D)$, gdzie $D \subset \mathbb{R}^n$ jest obszarem, natomiast dodatnia liczba rzeczywista p jest wykładnikiem całkowalności pierwszych (słabych) pochodnych funkcji określonych na D . Szersze omówienie teorii, w szczególności nierówności typu Sobolewa, można znaleźć w [1] [12]. W literaturze przedmiotu stała występująca w klasycznej nierówności Sobolewa [12] zależna jest od obszaru D , wymiaru n przestrzeni i wykładnika całkowalności p (np. [12] twierdzenia 4.10.1, 4.10.3, 4.10.5). W przypadku jednowymiarowym dla przestrzeni Sobolewa $W^{1,2}$ funkcji określonych na odcinku (a, b) w niniejszej pracy udało się wyznaczyć najlepszą stałą dla nierówności Sobolewa (twierdzenie 3).

F.1 Definicje

Niech $a < b$ będą dwiema liczbami rzeczywistymi. Niech γ będzie liczbą rzeczywistą z przedziału $(0, 1)$. Dla funkcji $f: (a, b) \rightarrow \mathbb{R}$, niech

$$\|f\|_{\infty} = \sup_{a < x < b} |f(x)| \quad (\text{norma supremum}),$$

$$\omega_{\gamma}(f) = \sup_{\substack{x, y \in (a, b) \\ x \neq y}} \frac{|f(x) - f(y)|}{|x - y|^{\gamma}} \quad (\gamma\text{-półnorma Höldera}),$$

$$\|f\|_{C^{0,\gamma}(a,b)} = \|f\|_{\infty} + \omega_{\gamma}(f) \quad (\gamma\text{-norma Höldera}).$$

Ponieważ w rozpatrywanym w pracy przypadku wymiar (odcinka) jest $n = 1$ oraz wykładnik całkowalności jest $p = 2$, więc $\gamma = \frac{1}{2}$ [12] [1].

Niech $\mathring{W}_{1,2}(a, b)$ oznacza przestrzeń różniczkowalnych funkcji rzeczywistych f takich, że

$\|f\|_{1,2} < \infty$, gdzie

$$\|f\|_{1,2} = (\|f\|_2^2 + \|f'\|_2^2)^{1/2} \quad ((1, 2)\text{-norma Sobolewa}).$$

oraz

$$\|f\|_2 = \left(\int_a^b |f(t)|^2 dt \right)^{1/2}, \quad \|f'\|_2 = \left(\int_a^b |f'(t)|^2 dt \right)^{1/2}.$$

F.2 Podstawowe nierówności

Twierdzenie 1. *Jeśli $f \in \mathring{W}^{1,2}(a, b)$, to*

$$\omega_{1/2}(f) \leq \|f'\|_2.$$

Dowód. Jeśli $x, y \in (a, b)$ spełniają $x < y$, to

$$f(y) - f(x) = \int_x^y f'(t) dt$$

i na mocy nierówności Cauchy'ego-Schwarza

$$\begin{aligned} |f(y) - f(x)| &\leq \int_x^y |f'(t)| dt \leq \left(\int_x^y |f'(t)|^2 dt \right)^{1/2} \left(\int_x^y 1^2 dt \right)^{1/2} \\ &= \sqrt{y-x} \|f'\|_2. \end{aligned}$$

Stąd, natychmiast, dla $x, y \in (a, b)$ takich, że $x \neq y$,

$$\frac{|f(x) - f(y)|}{\sqrt{|x-y|}} \leq \|f'\|_2. \quad (\text{F.1})$$

Biorąc supremum po wszystkich odpowiednich x, y otrzymujemy finalnie

$$\omega_{1/2}(f) \leq \|f'\|_2. \quad \square$$

Twierdzenie 2. *Jeśli $f \in \mathring{W}^{1,2}(a, b)$, to*

$$\|f\|_\infty \leq (b-a + (b-a)^{-1})^{1/2} \|f\|_{1,2}.$$

Dowód. Na mocy całkowego twierdzenia o wartości średniej istnieje $\xi \in (a, b)$ takie, że

$$f(\xi) = \frac{1}{b-a} \int_a^b f(t) dt. \quad (\text{F.2})$$

Jeśli $x \in (a, b)$, to, korzystając z (F.1) z y zastąpionym przez ξ ,

$$|f(x) - f(\xi)| \leq \sqrt{|x-\xi|} \|f'\|_2 \leq \sqrt{b-a} \|f'\|_2,$$

a stąd

$$|f(x)| \leq |f(\xi)| + \sqrt{b-a} \|f'\|_2. \quad (\text{F.3})$$

Z kolei, pamiętając o (F.2) i stosując ponownie nierówność Cauchy'ego-Schwarza, dostajemy

$$\begin{aligned} |f(\xi)| &\leq \frac{1}{b-a} \int_a^b |f(t)| dt \leq \frac{1}{b-a} \left(\int_a^b |f(t)|^2 dt \right)^{1/2} \left(\int_a^b 1^2 dt \right)^{1/2} \\ &= \frac{1}{b-a} \left(\int_a^b |f(t)|^2 dt \right)^{1/2} \sqrt{b-a} = \frac{1}{\sqrt{b-a}} \|f\|_2. \end{aligned}$$

Ostatnia nierówność wraz z (F.3) daje

$$|f(x)| \leq \sqrt{b-a} \|f'\|_2 + \frac{1}{\sqrt{b-a}} \|f\|_2,$$

a po wzięciu supremum po wszystkich x

$$\|f\|_\infty \leq \sqrt{b-a} \|f'\|_2 + \frac{1}{\sqrt{b-a}} \|f\|_2. \quad (\text{F.4})$$

Wykorzystując nierówność Cauchy'ego-Schwarza raz jeszcze otrzymujemy finalnie

$$\begin{aligned} \|f\|_\infty &\leq \left((\sqrt{b-a})^2 + \frac{1}{(\sqrt{b-a})^2} \right)^{1/2} (\|f'\|_2^2 + \|f\|_2^2)^{1/2} \\ &= (b-a + (b-a)^{-1})^{1/2} \|f\|_{1,2}. \quad \square \end{aligned}$$

Twierdzenie 3. *Jeśli $f \in \dot{W}^{1,2}(a, b)$, to*

$$\|f\|_{C^{0,1/2}(a,b)} \leq \left((\sqrt{b-a} + 1)^2 + (b-a)^{-1} \right)^{1/2} \|f\|_{1,2}.$$

Dowód. Z twierdzenia 1 oraz nierówności (F.4) otrzymano

$$\|f\|_{C^{0,1/2}(a,b)} \leq (\sqrt{b-a} + 1) \|f'\|_2 + \frac{1}{\sqrt{b-a}} \|f\|_2.$$

Teraz nierówność Cauchy'ego-Schwarza daje

$$\begin{aligned} \|f\|_{C^{0,1/2}(a,b)} &\leq \left((\sqrt{b-a} + 1)^2 + \frac{1}{(\sqrt{b-a})^2} \right)^{1/2} (\|f'\|_2^2 + \|f\|_2^2)^{1/2} \\ &= \left((\sqrt{b-a} + 1)^2 + (b-a)^{-1} \right)^{1/2} \|f\|_{1,2}. \quad \square \end{aligned}$$

Twierdzenie 4. *Jeśli f jest ciągłą funkcją na $[a, b]$ i różniczkowalną na (a, b) taką, że $f(a) = 0$ oraz $\|f'\|_2 < \infty$, to*

$$\|f\|_{C^{0,1/2}(a,b)} \leq (1 + \sqrt{b-a}) \|f'\|_2.$$

Dowód. Wraz z warunkiem $f(a) = 0$ nierówność (F.1) z $x = a$ implikuje, że

$$|f(y)| \leq \sqrt{y-a} \|f'\|_2$$

dla każdego $y \in (a, b)$. Stąd $|f(y)| \leq \sqrt{b-a} \|f'\|_2$ dla każdego $y \in (a, b)$, a dalej

$$\|f\|_\infty \leq \sqrt{b-a} \|f'\|_2.$$

Z drugiej strony, na mocy Twierdzenia 1 mamy

$$\omega_{1/2}(f) \leq \|f'\|_2.$$

□

I tak

$$\|f\|_{C^{0,1/2}(a,b)} \leq \|f'\|_2 + \sqrt{b-a} \|f'\|_2 = (1 + \sqrt{b-a}) \|f'\|_2.$$

Bibliografia

- [1] R. Adams, J. Fournier, Sobolev Spaces, 2003, Academic Press
- [2] S. Bernstein, Démonstration du théorème de Weierstrass fondée sur le calcul des probabilités, *Communications of the Kharkov Mathematical Society*, Volume XIII, 1912/13, p. 1-2.
- [3] J. F. Blinn, Models of light reflection for computer synthesized pictures. *Proc. 4th annual conference on computer graphics and interactive techniques*: 192–198. 1977
- [4] W. Böhm G. Farin, J. Kahmann, A survey of curve and surface methods in CAGD, *Computer Aided Geometric Design* Volume 1, Issue 1, July 1984, Pages 1-60
- [5] E. Catmull, R. Rom, A class of local interpolating splines. In *Computer Aided Geometric Design*, Eds. Academic Press, New York, 1974, pp. 317–326
- [6] M. Dryja, J. Jankowska, M. Jankowski: Przegląd metod i algorytmów numerycznych część 2. WNT 1982.
- [7] P. Dutre, P. Bekaert, K. Bala, *Advanced Global Illumination* 2nd Edition, A K Peters/CRC Press 2006
- [8] G. Farin, J. Hoschek, M.-S. Kim, *Handbook of Computer Aided Geometric Design*, Elsevier, 2002
- [9] A.R. Forrest, Interactive Interpolation and Approximation by Bézier Polynomials. *The Computer Journal*. 1972, vol.15 (1), ss. 71–79.
- [10] J. Gregory, *Game Engine Architecture*, 2nd Edition, A K Peters/CRC Press 2014
- [11] D. Holden, T. Komura, J. Saito, Phase-functioned neural networks for character control, *ACM Transactions on Graphics* Volume 36 Issue 4, July 2017 Article No. 42.
- [12] T. Iwaniec, G. Martin, *Geometric Function Theory and Non-linear Analysis*, 1st Edition, Clarendon Press, 2002

- [13] Ł. Izdebski, D. Sawicki: Easing functions in the new form based on Bézier Curves, *Proceedings of the International Conference on Computer Vision and Graphics (ICCVG 2016)*, Warszawa. September 19-21, 2016. LNCS Springer. LNCS 9972, 2016, pp. 37–48.
- [14] O. Johnston, F. Thomas: *The Illusion of Life: Disney Animation*. Disney Editions 1995
- [15] C. Jones, Chuck Amuck: *The Life and Times of an Animated Cartoonist*, 1st edition, Farrar, Straus and Giroux, New York, 1990, p. 180.
- [16] P. Kiciak, *Podstawy modelowania krzywych i powierzchni*, wydanie II, WNT 2005
- [17] D. H. U. Kochanek, R. H. Bartels, Interpolating splines with local tension, continuity, and bias control, *ACM SIGGRAPH 1984*, vol. 18, no. 3, pp. 33-41.
- [18] B. Koyama-Richard, *1000 lat historii mangi*, Wydawnictwo Naukowe PWN, 2008
- [19] L. Kowalski, A. Piskorek, *Analiza Wypukła*, 2013, Wydawnictwo Wojskowa Akademia Techniczna
- [20] F. Leja, *Teoria funkcji analitycznych*, Wydawnictwo Naukowe PWN, 1957
- [21] F. Luna, *Introduction to 3D Game Programming with DirectX 11*, Mercury Learning & Information 2012
- [22] N. Magnenat-Thalmann, D. Thalmann, *Computer Animation Theory and Practice*, Springer Japan 1990
- [23] J.E. Mardsen, L. Sirovich, Piecewise cubic interpolation, *Applied Mathematical Sciences*, vol 27. Article No. 3, pp. 40-48.
- [24] K. Maurin, *Metody przestrzeni Hilberta*, 1959, Warszawa Wydawnictwo Naukowe PWN
- [25] M. McShaffry, D. Graham, *Game Coding Complete*, 4th Edition, Cengage Learning PTR 2012
- [26] R. Parent, *Computer Animation: Algorithms and Techniques*, 3rd Edition, Morgan Kaufmann 2012
- [27] R. Penner, *Robert Penner's Programming Macromedia Flash MX*, 1st Edition, McGraw-Hill/OsborneMedia 2002
- [28] M. Pharr, W. Jakob, G. Humphreys, *Physically Based Rendering: From Theory to Implementation* 3rd Edition, Morgan Kaufmann 2016
- [29] D. Rosen, *Animation Bootcamp: An Indie Approach to Procedural Animation*, GDC 2014

- [30] W. Rudin, *Analiza funkcjonalna*, Warszawa, wydanie 2, 2002, Wydawnictwo Naukowe PWN
- [31] W. Rudin, *Analiza rzeczywista i zespolona*, Warszawa, 2, 2009, Wydawnictwo Naukowe PWN
- [32] W. Rudin, *Podstawy analizy matematycznej*, Warszawa, wydanie 2, 1982, Wydawnictwo Naukowe PWN
- [33] C. Runge, Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten, *Zeitschrift für Mathematik und Physik*, 46: 224–243. 1901
- [34] S. Saks, A. Zygmunt, *Funkcje analityczne*, Warszawa, 1959, Wydawnictwo Naukowe PWN
- [35] G. Seller, R. S. Wright Jr., N. Haemel, *OpenGL Superbible: Comprehensive Tutorial and Reference*, 7 edition, Addison-Wesley Professional 2015
- [36] G. Sellers, J. Kessenich, *Vulkan Programming Guide: The Official Guide to Learning Vulkan*, 1 edition, Addison-Wesley Professional 2016
- [37] P. Shirley, *Realistic Ray Tracing*, 2nd Edition, Routledge 2003
- [38] P. Shirley, *Fundamentals of Computer Graphics 3rd Edition*, Taylor&Francis Ltd. 2010
- [39] S. Soboleff, Sur un th éor’eme d’analyse fonctionnelle, *Rec. Math. [Mat. Sbornik] N.S.*, 1938, Volume 4(46), Number 3, 471–497
- [40] M. H. Stone. The generalized Weierstrass approximation theorem. *Math. Mag.* 21, (1948). 167-184, 237-254.
- [41] J. Szabatin, *Podstawy teorii sygnałów*, Wydawnictwa Komunikacji i Łączności, 2003
- [42] K. Tatarkiewicz, *Rachunek wariacyjny cz. I*, Warszawa, 1969, Wydawnictwo Naukowe PWN
- [43] A. Watt, M. Watt, *Advanced Animation and Rendering Techniques*, Addison-Wesley Professional 1992
- [44] C. Yuksel, S. Schaefer, J. Keyser: On the Parameterization of Catmull-Rom Curves. *Proc. of the SPM’09, SIAM/ACM Joint Conference on Geometric and Physical Modeling*. San Francisco October 05-08, 2009, pp.47-53.
- [45] *animacja*, Encyklopedia powszechna PWN, 1999, encyklopedia.pwn.pl/encyklopedia/animacja.html, /dostęp 19.05.2019/

- [46] Monte Carlo Path Tracing, www.graphics.stanford.edu/courses/cs348b-01/course29.hanrahan.pdf, /dostęp 19.05.2019/
- [47] COLLADA – Digital Asset Schema Release 1.5.0, www.khronos.org/files/collada_spec_1_5.pdf, /dostęp 19.05.2019/
- [48] Open Game Engine Exchange Specification, opengex.org/OpenGEX.1.1.2.pdf, /dostęp 19.05.2019/
- [49] IEEE Std 754™-2008 (Revision of IEEE Std 754-1985) IEEE Standard for Floating-Point Arithmetic, www.dsc.ufcg.edu.br/~cnum/modulos/Modulo2/IEEE754_2008.pdf, /dostęp 19.05.2019/
- [50] Detection of first and second order motion, papers.nips.cc/paper/1371-detection-of-first-and-second-order-motion.pdf, /dostęp 19.05.2019/
- [51] DeepMotion: AI Driven Motion for Games, www.80.lv/articles/deepmotion-ai-driven-motion-for-games/, /dostęp 19.05.2019/
- [52] NaturalMotion's Euphoria, www.youtube.com/watch?v=HauN98naZ9U, /dostęp 19.05.2019/
- [53] MOTION CAPTURE Przechwytywanie ruchu, www.mocap.pl/blog/motion-capture-przechwytywanie-ruchu, /dostęp 19.05.2019/
- [54] Physics-Based Computer Animation, lgg.epfl.ch/research_physicsbased_animation.php, /dostęp 19.05.2019/
- [55] Cycles Open Source Production Rendering, www.cycles-renderer.org/, /dostęp 19.05.2019/
- [56] VESA® Adds 'Adaptive-Sync' to Popular DisplayPort™ Video Standard, vesa.org/featured-articles/vesa-adds-adaptive-sync-to-popular-displayport-video-standard, /dostęp 19.05.2019/
- [57] Animator QML Type, doc.qt.io/qt-5/qml-qtquick-animator.html, /dostęp 19.05.2019/
- [58] Easing Functions, docs.microsoft.com/en-us/dotnet/framework/wpf/graphics-multimedia/easing-functions, /dostęp 19.05.2019/
- [59] Cg Programming/Unity/Hermite Curves, en.wikibooks.org/wiki/Cg_Programming/Unity/Hermite_Curves, /dostęp 19.05.2019/

- [60] Bézier interpolation source code, developer.blender.org/diffusion/B/browse/master/source/blender/blenkernel/intern/fcurve.c, /dostęp 19.05.2019/
- [61] FBX SDK Programmer's Guide, docs.autodesk.com/FBX/2014/ENU/FBX-SDK-Documentation/, /dostęp 19.05.2019/
- [62] Lagrange Interpolating Polynomial, mathworld.wolfram.com/LagrangeInterpolatingPolynomial.html, /dostęp 19.05.2019/
- [63] Bézier curve, https://www.encyclopediaofmath.org/index.php/Bézier_curve, /dostęp 19.05.2019/
- [64] Unreal Engine 4 Documentation, docs.unrealengine.com/en-us/, /dostęp 19.05.2019/
- [65] Animation Sequences, docs.unrealengine.com/en-us/Engine/Animation/Sequences, /dostęp 19.05.2019/
- [66] Johansson: Motion Perception part 1, www.youtube.com/watch?v=1F5ICP9SYLU, /dostęp 19.05.2019/
- [67] Johansson: Motion Perception part 2, www.youtube.com/watch?v=KT89CQ2nRpo, /dostęp 19.05.2019/
- [68] Known Maximum Errors in Math Functions, www.gnu.org/software/libc/manual/html_node/Errors-in-Math-Functions.html#Errors-in-Math-Functions, /dostęp 19.05.2019/
- [69] Intel overstates FPU accuracy, www.notabs.org/fpuaccuracy, /dostęp 19.05.2019/
- [70] www.alanzucconi.com/2017/04/17/procedural-animations/, /dostęp 19.05.2019/
- [71] knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2016/ENU/3DSMax/files/GUID-10AC3D82-523D-4670-8240-D6E3AC79F0BF-htm.html, /dostęp 19.05.2019/
- [72] IK Solvers, knowledge.autodesk.com/support/3ds-max/learn-explore/caas/CloudHelp/cloudhelp/2017/ENU/3DSMax/files/GUID-AE4A0089-95F5-4199-A853-ABB8E0DB3439-htm.html, /dostęp 19.05.2019/