



**AGH**

AKADEMIA

GÓRNICZO-HUTNICZA

STANISŁAWA

STASZICA

W

KRAKOWIE

**Wydział Fizyki i Informatyki Stosowanej**

---

---

## **Praca magisterska**

**Łukasz Wędzony**

kierunek studiów: **fizyka techniczna**

specjalność: **fizyka komputerowa**

**Marek Wywijas**

kierunek studiów: **informatyka stosowana**

specjalność: **informatyka w nauce i technice**

# **Model szybowca w grafice 3D z uwzględnieniem fizyki lotu**

**Opiekun: dr hab. inż. Tadeusz Szuba; prof. nzw. AGH**

**Kraków, marzec 2013**

Oświadczam, świadomy(-a) odpowiedzialności karnej za poświadczenie nieprawdy, że niniejszą pracę dyplomową wykonałem(-am) osobiście i samodzielnie i nie korzystałem(-am) ze źródeł innych niż wymienione w pracy.

.....

(czytelny podpis)

.....

(czytelny podpis)

Kraków, 17 marca 2013

Tematyka pracy magisterskiej i praktyki dyplomowej Łukasza Wędzonego studenta V roku studiów kierunku fizyka techniczna, specjalności fizyka komputerowa oraz Marka Wywijasa studenta V roku studiów kierunku informatyka stosowana, specjalności informatyka w nauce i technice


Opiekun pracy: dr hab. inż. Tadeusz Szuba; prof. nzw. AGH

Recenzenci pracy:

Miejsce praktyki dyplomowej: WFiIS AGH, Kraków

Program pracy magisterskiej i praktyki dyplomowej

# Spis treści

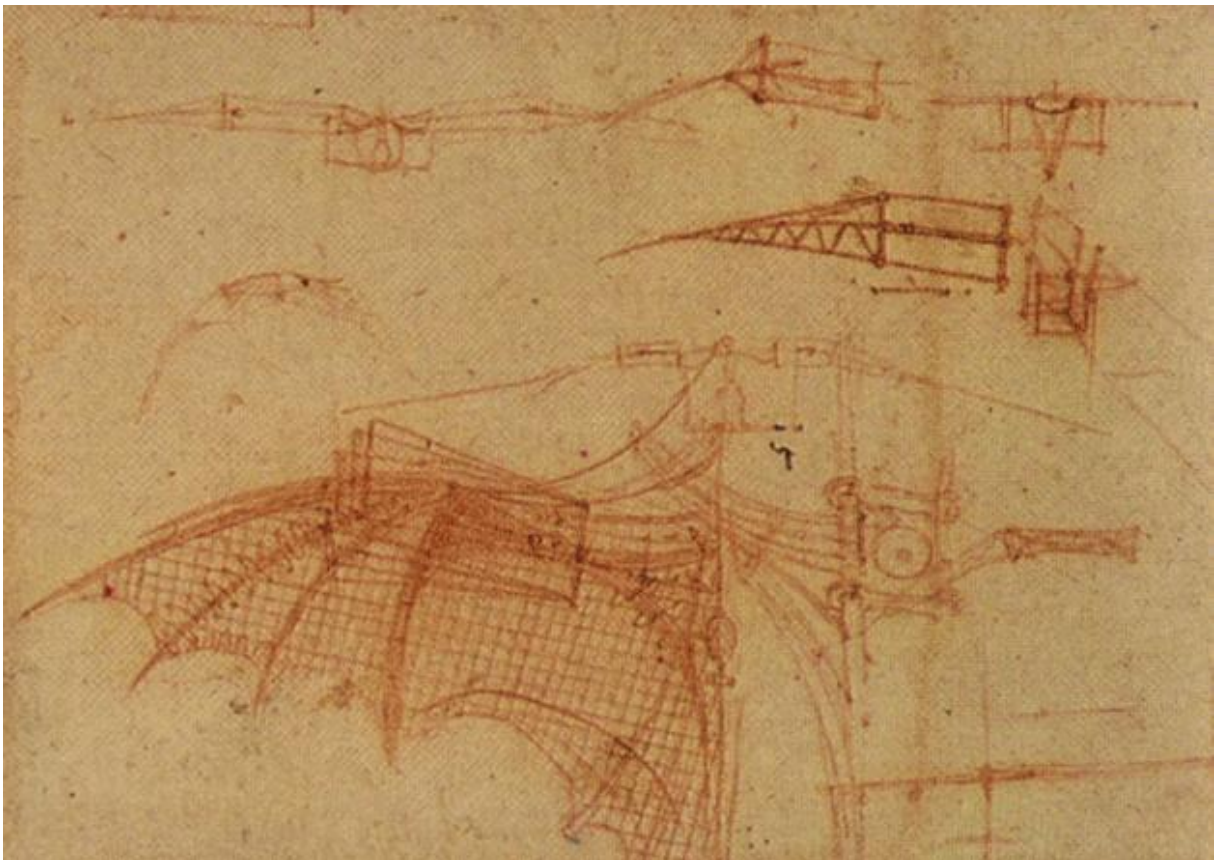
1	Wstęp .....	6
2	Teoria.....	9
2.1	Siły działające na szybowiec .....	10
2.1.1	Siła nośna.....	12
2.1.2	Opór..... 	20
2.2	Konstrukcja szybowca .....	32
2.2.1	Powierzchnie nośne.....	33
2.2.2	Porównanie właściwości profili skrzydeł .....	39
2.2.3	Elementy mechanizacji skrzydła .....	48
3	Konstrukcja modelu.....	55
3.1	Model szybowca .....	56
3.1.1	Geometria modelu .....	56
3.1.2	Oświetlenie i materiały.....	66
3.1.3	Detale .....	70
3.1.4	Inne modele.....	72
3.2	Model otoczenia.....	75
4	Symulacja.....	88
4.1	Model fizyczny .....	88
4.1.1	Obliczenia dotyczące sił i ruchu – metoda Eulera, Tensory i Kwaterniony. ....	88
4.1.2	Schemat programu .....	92
4.2	Środowisko programistyczne .....	94
4.3	Opis programu.....	102
4.3.1	Opis panelu sterowania, wejścia i wyjścia skryptu.....	102
4.3.2	Obliczenia przed startem symulacji.....	103

4.3.3	Warunki atmosferyczne .....	104
4.3.4	Opis warunków początkowych.....	106
4.3.5	Krok symulacji.....	106
4.4	Przykładowe wyniki działania .....	108
5	Podsumowanie .....	110
6	Dodatki .....	112
6.1	Dodatek A - doświadczenie z profilem skrzydła .....	112
6.2	Dodatek B – kamera śledząca.....	117
6.3	Dodatek C – chmury .....	118
7	Spis ilustracji.....	122
8	Bibliografia.....	126



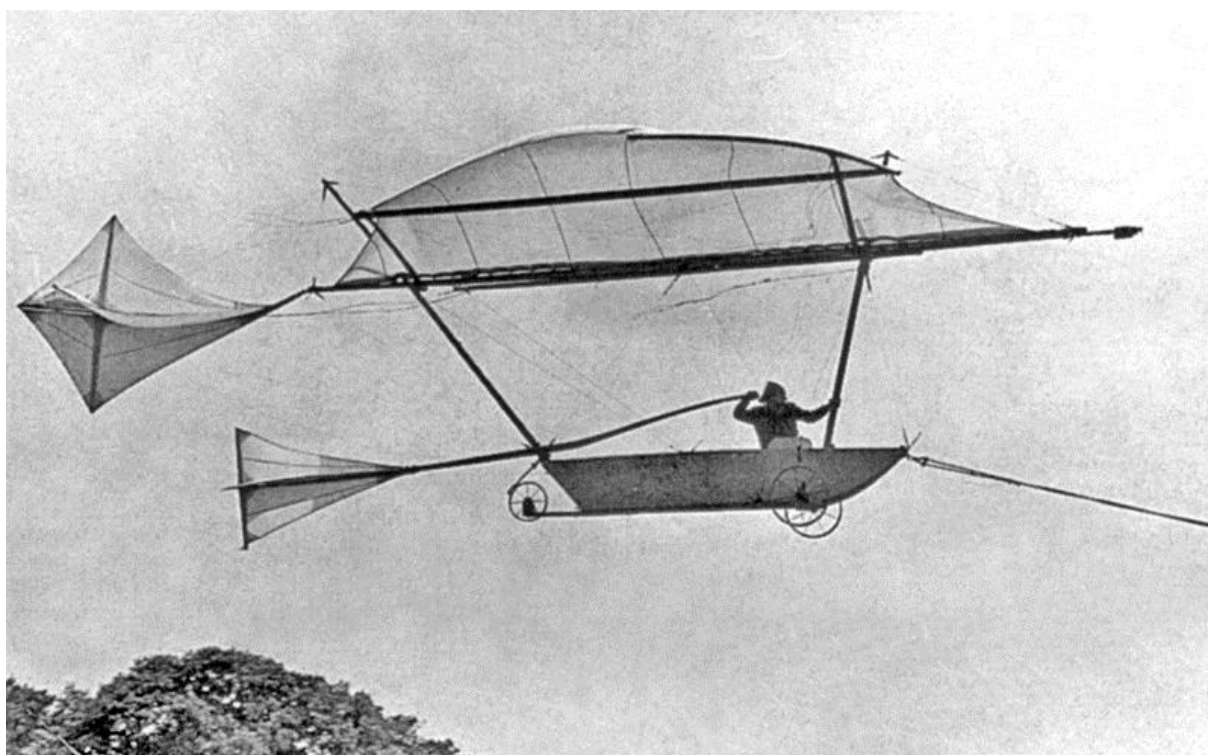
# 1 Wstęp

Ludzkość od zarania dziejów fascynowała i marzyła o lataniu. Rozważania na temat statku powietrznego mogącego ponieść człowieka w przestworza sięgają jeszcze czasów Leonarda Da Vinci, który na przełomie XV i XVI wieku napisał „Traktat o locie ptaków”. Dzieło to zawiera studia nad lotem ptaków, omawiając zagadnienia takie jak mechanika lotu, w tym wpływ oporów powietrza oraz wiatru i prądów powietrznych, a także rozważania na temat anatomii tych stworzeń. Następnie w swojej pracy rozważania te Leonardo Da Vinci przeradza w szkic konstrukcji maszyny latającej. Maszyna ta nigdy nie została jednak wybudowana.



Rys. 1.1 Plany maszyny latającej Leonarda Da Vinci

Człowiekiem, który skonstruował pierwszy na świecie szybowiec zdolny do uniesienia człowieka był Brytyjczyk Sir George Cayley, który w 1852 roku wynalazł „Governable Parachute”. Od tamtego czasu rozpoczęła się ekspansja w kierunku podboju przestworzy, której najbardziej przełomowym momentem było wyoszarzenie przez braci Wright statku powietrznego w silnik spalinowy. Maszyna tego typu odbyła swój pierwszy, lot w 1903 roku. Trzy lata później bracia Wright złożyli wniosek o patent na swój sposób sterowania maszynami latającymi. Wydarzenie to spowodowało rozwój lotnictwa ukierunkowanego na konstrukcje maszyn wyposażonych w silnik jednocześnie spychając szybownictwo do sfery sportu, w której pozostaje po dziś dzień.



Rys. 1.2 Replika szybowca Georga Cayley'a w trakcie lotu (rok 1973)

Historia grafiki komputerowej sięga zaledwie lat pięćdziesiątych XX wieku. W początkach istnienia ogromne koszty powodowały, że głównie stosowana ona była na potrzeby militarne. Szybki rozwój technologii komputerowej spowodował jednak, że grafika komputerowa stopniowo zaczynała być stosowana w wielu innych dziedzinach. W dzisiejszych czasach wachlarz zagadnień, do których opisu wykorzystuje się grafikę komputerową jest tak rozległy, iż można ją określić mianem metajęzyka. W naszej pracy postaramy się dostarczyć przykładów tej teorii poprzez przedstawienie szybownictwa językiem grafiki komputerowej. W tym celu potrzebna jest jednak bardzo dobra znajomość natury omawianych zjawisk. Dlatego też w pierwszych rozdziałach pracy skoncentrujemy się na zgłębieniu tajników zjawisk fizycznych kryjących się za możliwością wznoszenia się człowieka w przestworza. W dalszej części rozważań teoretycznych przyjrzymy się także w jaki sposób skonstruowane są szybowce. Następnie przejdziemy do tworzenia własnego modelu graficznego, którym postaramy się w jak najlepszy sposób oddać zjawisko lotu. W

modelu naszym zawrzemy zarówno graficzne przedstawienie szybowców jak i otoczenia, w którym będą się one poruszać. Modele szybowców wyposażymy następnie w poprawnie prezentującą się od strony graficznej fizykę lotu.

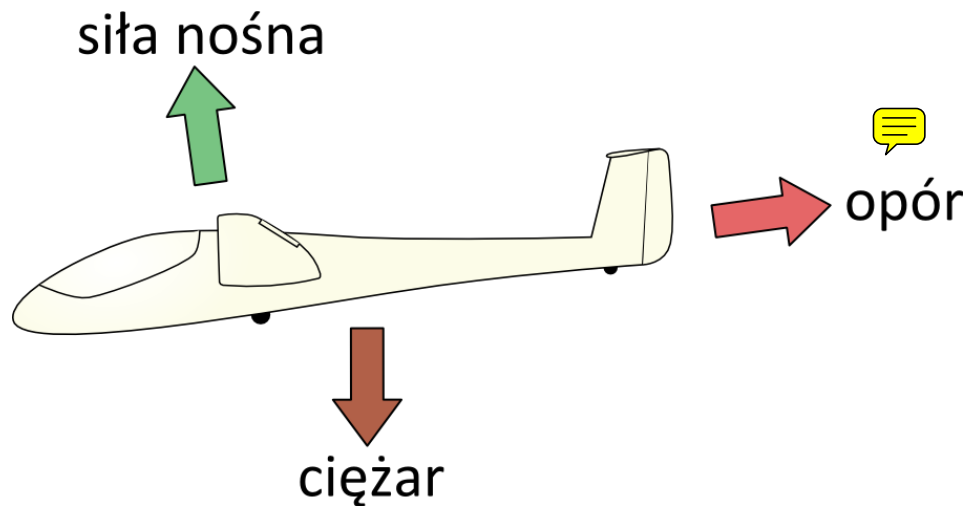


## 2 Teoria

W rozdziale pracy poświęconym teorii, przedstawimy siły i zjawiska oddziałujące na latający statek powietrzny, a następnie szczególnie przyjrzymy się ich właściwościom w przypadku szybowca. Pokażemy także ich źródło oraz sposób ilościowego wyznaczenia wartości charakteryzujących powierzchnie nośne. Rozpocznemy od ogólnego rozrysowania sił, aby następnie szczegółowo wyprowadzić potrzebne zależności wykorzystane przy budowie modelu fizycznego. Samo zjawisko powstawania sił aerodynamicznych, a więc takich które wynikają z opływu powietrza wokół ciała i niejednostajnego rozkładu ciśnienia na jego powierzchni daje się wytłumaczyć i wyjaśnić na podstawie określonych zjawisk i praw fizycznych wychodzących z zasady zachowania energii. Pomimo tego analityczne wyznaczenie dokładnych wartości powstających sił jest możliwe tylko dla nielicznych przypadków używanych w rzeczywistości powierzchni nośnych. Możemy tego dokonać przez zastosowanie przekształceń konforemnych, jednak takie wyliczenia dadzą nam jedynie wyniki opisujące model otoczony idealną cieczą. Aby poznać rzeczywiste wartości rozkładu ciśnień, prędkości i sił przepływającej cieczy (znając rozkład ciśnień jesteśmy w stanie wyliczyć pozostałe wielkości) można podążać dwoma drogami. Pierwszą z nich jest rozwiązanie równań Naviera-Stokesa. Druga wymaga natomiast przeprowadzania doświadczeń (na przykład w tunelu aerodynamicznym). Równania N-S, daje się rozwiązać analitycznie jedynie dla najprostszyc przypadków. W rzeczywistości stosuje się do tego skomplikowane symulacje numeryczne. Pomimo szybkiego rozwoju komputerów i wzrostu dostępnej prędkości obliczeniowej wciąż trzeba stosować rozmaite przybliżenia, stąd zagadnienie rozwiązywania równań N-S jest bardzo złożone i samo w sobie jest tematem prac magisterskich i doktorskich. Z tego też powodu historycznie pierwszym sposobem wyznaczenia wartości parametrów niezbędnych przy projektowaniu samolotów czy też ogólnie powierzchni nośnych było przeprowadzanie doświadczeń. Obecnie ogólnie przyjętym sposobem wprowadzania nowych rozwiązań jest więc tworzenie i wstępne testowanie elementu w komputerze, a następnie testowanie w tunelu aerodynamicznym zarówno pod kątem własności aerodynamicznych jak i materiałowych – wytrzymałości czy elastyczności. Zastosowanie nowoczesnych komputerów i technik programowania pozwoliło stosować ten test jako ostateczny egzamin - najczęściej potwierdzający wyniki symulacji, a nie jako element procesu rozwojowego, w którym to testy trzeba było wielokrotnie powtarzać. Opis prostego doświadczenia w tunelu aerodynamicznym przedstawiony został na końcu pracy (Dodatek A - doświadczenie z profilem skrzydła).

## 2.1 Siły działające na szybowiec

Na szybowiec znajdujący się w powietrzu oddziałuje siła grawitacji związana z jego ciężarem. Zgodnie z pierwszym prawem Newtona, aby szybowiec utrzymał się w powietrzu potrzebuje wytworzyć siłę o przeciwnym zwrocie. Siła ta, zwana siłą aerodynamiczną, wytwarzana jest dzięki opływaniu w trakcie lotu mas powietrza znajdujących się wokół szybowca. Siła aerodynamiczna składa się z dwóch czynników. Pierwszym z nich jest siła nośna wytwarzana przez skrzydła szybowca. Jest to siła prostopadła do kierunku opływającego powietrza skierowana w górę, co powoduje że w dużym stopniu równoważy siłę grawitacji co pozwala szybowcowi pozostać w powietrzu. Drugą składową jest opór wytwarzany poprzez lecący statek powietrzny o zwrocie zgodnym z poruszającymi się wokół niego masami powietrza (Rys. 2.1). Rozkład powyższych sił w przypadku dwuwymiarowym zapisujemy przy użyciu równań ruchu (Równanie 1).



Rys. 2.1 Siły działające na szybowiec

$$\begin{aligned} m\ddot{x} &= D \cos\alpha + L \sin\alpha \\ m\ddot{y} &= D \sin\alpha + L \cos\alpha - mg \end{aligned} \quad \text{a)}$$

A więc,

$$\begin{cases} \dot{x} = \frac{D}{m} \cos\alpha t - \frac{L}{m} \sin\alpha t \\ \dot{y} = \frac{D}{m} \sin\alpha t + \frac{L}{m} \cos\alpha t - gt \end{cases} \quad \text{b)}$$

Równanie 1 Równania ruchu w dwóch wymiarach

gdzie:

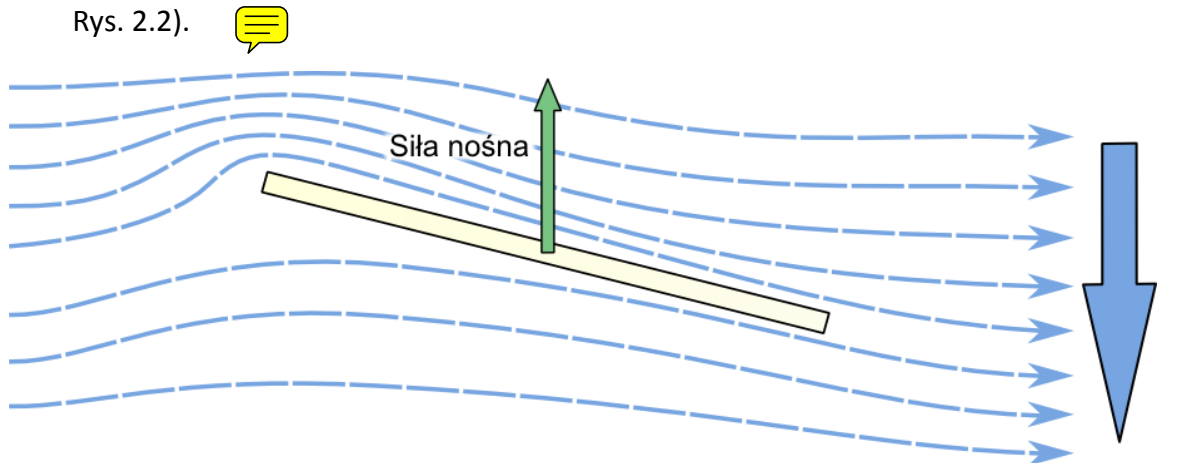
- $L$  – siła nośna
- $D$  – opór
- $g$  – przyspieszenie ziemskie
- $m$  – masa szybowca
- $t$  – czas
- $\alpha$  - kąt nachylenia toru lotu szybowca

Aby jednak móc powyższe równania rozwiązać, należy przede wszystkim wyznaczyć siły  $L$  i  $D$ . Właśnie ten temat zajmie główną część rozdziału dotyczącego teorii.

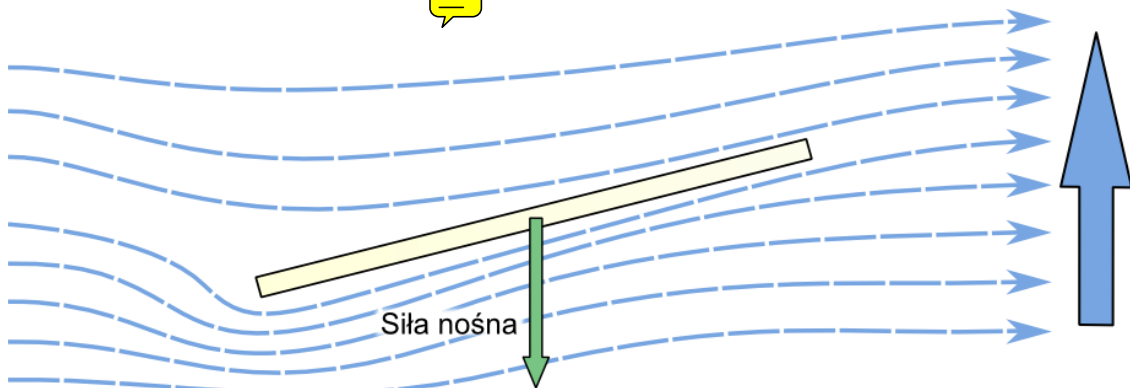
## 2.1.1 Siła nośna

Dowolne ciało, które poruszając się w ziemskiej atmosferze zmienia kierunek ruchu opływających go mas powietrza wytwarza siłę nośną. Związane jest to z trzecią zasadą dynamiki Newtona, z której wynika, że akcji (zmianie kierunku mas powietrza) towarzyszyć musi reakcja (siła działająca na poruszające się ciało). Wartość i kierunek wytwarzanej siły zależy od własności fizycznych danego ciała. Przykładem ciała, które poruszając się w atmosferze wytwarza siłę nośną może być zwykły płaski kawałek drewna (

Rys. 2.2).



- a) ciało ustawione w tej pozycji spycha opływające powietrze w dół przez to wytworzona siła reakcji skierowana jest w górę



- b) w tej pozycji powietrze spychane jest do góry przez co wytworzona siła reakcji skierowana jest w dół



- c) ciało ustawione w tej pozycji nie wytwarza siły nośnej ponieważ nie zmienia kierunku ruchu powietrza (lub zmiany te się równoważą)

Rys. 2.2 Przykładowe ciało poruszające się w ziemskiej atmosferze (drugi czynnik siły aerodynamicznej – opór został tutaj pominięty. Zostanie on omówiony w rozdziale 2.1.2)



Wytwarzanie siły nośnej związane jest więc ze zmianą pędu opływającego powietrza. Powietrze to posiada energię kinetyczną:

$$E_k = \frac{mv^2}{2}$$

**Równanie 2 Energia** kinetyczna spychanego powietrza

gdzie:

- $m$  – masa spychanego powietrza
- $v$  – jego prędkość



Przyjmijmy stałą prędkość powietrza. W takiej sytuacji energia jest łatwa do wyznaczenia, ponieważ pozostaje tylko określić masę powietrza, którego pęd ulega zmianie. Przyjmijmy, że masa ta jest iloczynem objętości i gęstości:

$$E_k = \frac{\rho V v^2}{2}$$

**Równanie 3 Szacowana** energia spychanego powietrza

tutaj:

- $\rho$  – gęstość powietrza
- $V$  - objętość powietrza reagującego z ciałem

W tej sytuacji pozostaje tylko określić objętość powietrza odbijanego przez obiekt. Jednak wpływ ten zależy od wielu czynników takich jak:

- Rozmiar ciała
- Kształt powierzchni
- Nachylenie

Powoduje to, że nie jesteśmy w stanie jednoznacznie określić objętości oddziaływania. Jedyne co możemy z całą pewnością wyznaczyć to powierzchnia ciała poruszającego się w powietrzu. Za pozostałą niewiadomą przyjmiemy czynnik i oznaczmy go jako  $c$ . W ten sposób otrzymujemy równanie siły aerodynamicznej (całości powstałej siły, więc uwzględniamy też opór) wynikającej z oddziaływania ciała z masą powietrza:

$$F_a = \frac{C}{2} \rho S v^2$$

Równanie 4 Powstała siła aerodynamiczna

gdzie:

- $C$  – współczynnik aerodynamiczny
- $S$  – powierzchnia ciała



Wprowadzimy teraz dwie bezwymiarowe wielkości: współczynnik siły nośnej  $C_l$  oraz współczynnik oporu  $C_d$ .

$$C_l = \frac{2L}{\rho |V_\infty|^2 c}$$

Równanie 5 Współczynnik siły nośnej

$$C_d = \frac{2D}{\rho |V_\infty|^2 c}$$

Równanie 6 Współczynnik oporu

Gdzie  $D$  i  $L$  to opór i siła nośna,  $\rho$  jest gęstością cieczy,  $|V_\infty|$  to wartość prędkości strumienia niezaburzonej cieczy (w oddali od profilu), a  $c$  jest cięciwą.

Powodem zastosowania powyższych wielkości jest możliwość opisywania i porównywania różnych profili skrzydeł niezależnie od wielkości modelu, gęstości czy też prędkości cieczy, przy jakiej jest on badany. Zwykle interesująca jest przede wszystkim zależność wydajności profilu od kąta natarcia. Takie właśnie dane w postaci wykresów są najczęściej wykorzystywane do opisywania profili w bazach danych. Odzyskanie potem konkretnej wartości siły z takiego wykresu jest przy zastosowaniu powyższych wzorów bardzo łatwe.

Siłę aerodynamiczną możemy więc rozbić na dwie składowe jak na Rys. 2.1, otrzymamy wtedy:

$$L = \frac{C_L}{2} \rho S v^2$$

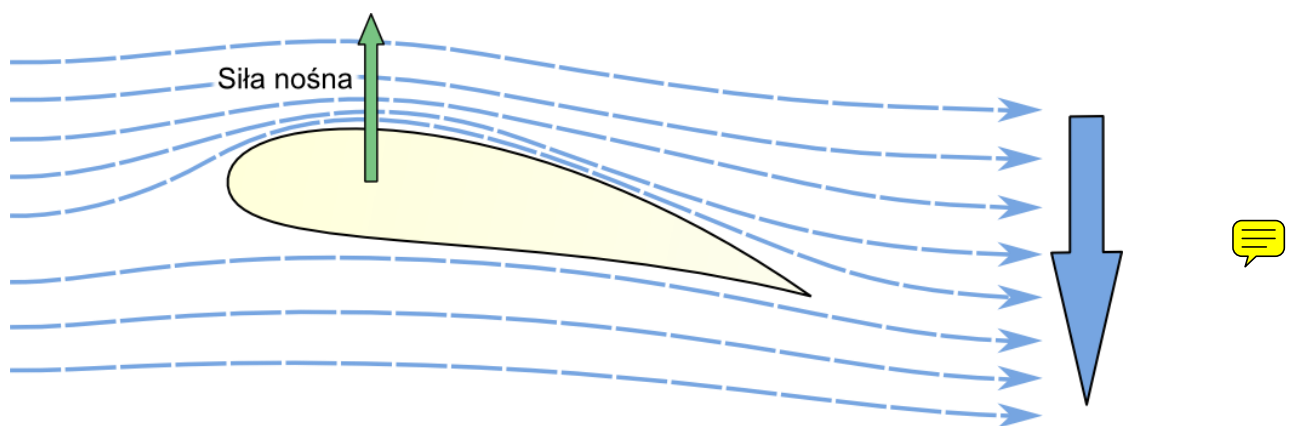
a) siła nośna

$$D = \frac{C_D}{2} \rho S v^2$$

b) opór

Równanie 7 Rozkład siły aerodynamicznej na składowe

Powyższy zapis jest więc o tyle nietypowy, że uzależnia siłę od wtórnego współczynnika. Jest jednak uzasadniony i wynika ze wspomnianego we wstępie do teorii sposobu podejścia do problemu wyznaczania działających sił i danych jakimi będziemy dysponować przed rozwiązaniem problemu.



Rys. 2.3 Przykładowy profil lotniczy poruszający się w powietrzu

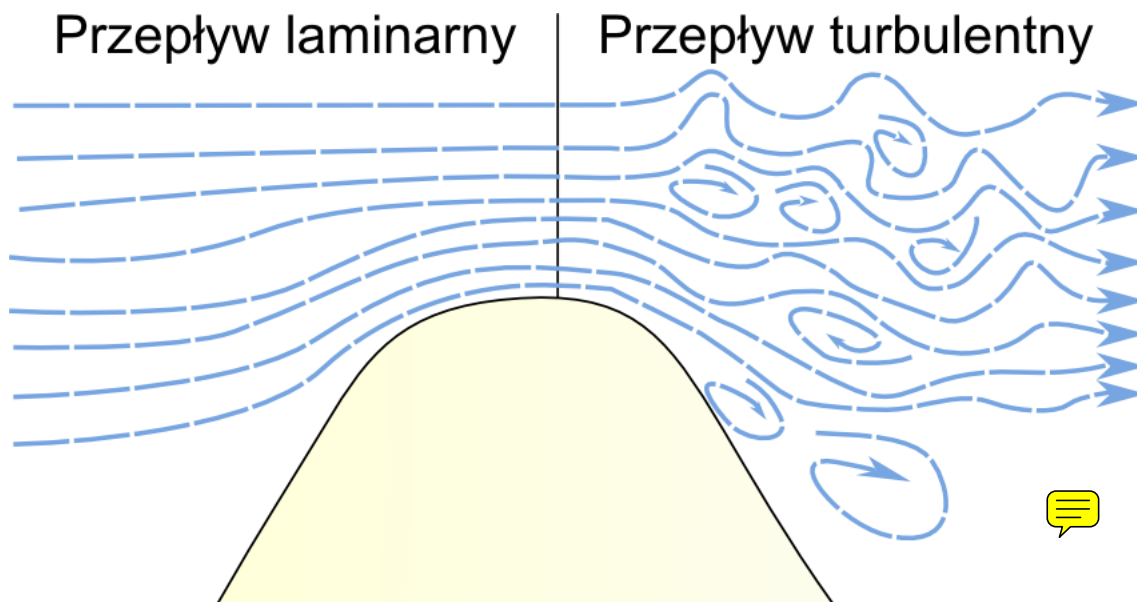
W przypadku szybowca na powstawanie siły nośnej mają wpływ wszystkie elementy konstrukcji, lecz zdecydowanie największy wkład wnoszą skrzydła, które są doskonałym przykładem powierzchni nośnych, czyli powierzchni wyprofilowanych w ten sposób, żeby wytwarzać dużą siłę nośną przy jednoczesnej minimalizacji powstającego oporu (więcej na ten temat powiemy sobie w rozdziale 2.2.1). Dobrą nośność skrzydeł zapewnia specjalistyczny profil lotniczy, czyli przekrój skrzydła specjalnie zaprojektowany tak żeby spełniać narzucone wymagania nośności (Rys. 2.3).

### 2.1.1.1 Ciecz Idealna, równanie Bernoulliego

Kwestia oddziaływań w układzie ciała poruszającego się w płynie jest jednak dużo bardziej skomplikowana. Zaczniemy od uzupełnienia naszych dotychczasowych rozważań o pojęcie ciśnienia. Ciśnienie płynu jest to siła, z jaką naciska on na jednostkę powierzchni ciała. Wyróżniamy dwa rodzaje ciśnienia:

- Ciśnienie statyczne - siła, z jaką płyn pozostający w spoczynku wywiera nacisk na jednostkę powierzchni ciała. Powstanie ciśnienia statycznego związane jest z tym, że pojedyncze cząsteczki płynu pozostają w ciągłym ruchu, co powoduje, że te znajdujące się w pobliżu zanurzonego ciała kolidują z nim i ulegają odbiciu, co powoduje zmianę ich pędu. Z trzeciej zasady dynamiki powstaje więc siła reakcji. Ponieważ jednak cząstki oddziałują we wszystkich kierunkach równocześnie (płyn pozostaje w spoczynku) powstała siła nie posiada kierunku. Ciśnienie statyczne jest więc wartością skalarną
- Ciśnienie dynamiczne – siła wytwarzana przez płyn pozostający w ruchu, o kierunku zgodnym z ruchem płynu

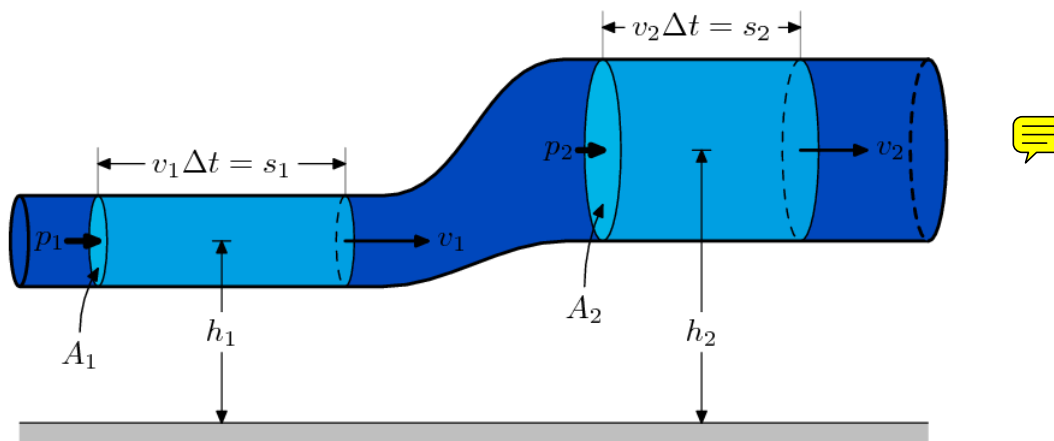
W celu przeprowadzenia dalszych rozważań przyjmijmy model powietrza jako płynu z pewnym uproszczeniem. Przyjmijmy, że powietrze jest płynem doskonałym – a więc takim, który jest nieściśliwy i nie posiada lepkości. Jeśli chodzi o założenie dotyczące ściśliwości, to powietrze spełnia je poniżej prędkości dźwięku, a również poniżej prędkości dźwięku znajduje się dostępny szybowcom zakres prędkości. Jeśli jednak chodzi o efekty związane z lepkością – istotnie występują one i będą uwzględnione również w naszym modelu, jednak dla przejrzystości prowadzonego rozumowania opiszemy je w rozdziale 2.1.2



Rys. 2.4 Schematyczne przedstawienie przepływu laminarnego oraz turbulentnego



Lepkość jest właściwością płynów, która dla modelu laminarnego przepływu określa przekazywanie pędu pomiędzy warstwami poruszającymi się z różnymi prędkościami. Warstwy są w mechanice płynów pojęciem hipotetycznym, ponieważ zmiana prędkości zachodzi w sposób ciągły. Aby jednak przyjąć powyższe określenie lepkości musimy mieć do czynienia z wspomnianym modelem laminarnym, czyli nieturbulentnym. Ruch turbulentny jest określeniem bardzo złożonego zachowania płynów, w którym pojawiają się wiry oraz inne struktury niekoherentne (Rys. 2.4). Dla przepływu turbulentnego nie istnieją dobre modele teoretyczne.



Ciecz płynąca w rurze przedostając się do jej szerszej części ( $A_2 > A_1$ ) zmniejsza swoją prędkość ( $v_2 < v_1$ ) co wiąże się ze wzrostem ciśnienia w tej części rury ( $p_2 > p_1$ ).

Rys. 2.5 Demonstracja praktycznego zastosowania równania Bernoulliego

Układ ciała poruszającego się w płynie musi spełniać zasadę zachowania energii, którą przy przyjęciu opisanych założeń:

- płyn jest nieściśliwy
- płyn nie jest lepki
- przepływ jest stacjonarny (niezmienny w czasie) i laminarny

możemy wyrazić pod postacią równania Bernoulliego.

W 18-tym wieku szwajcarski matematyk i fizyk Daniel Bernoulli zdefiniował jedno z podstawowych równań, pozwalających zrozumieć zjawiska mechaniki w płynach. Równanie to jako zasada zachowania energii może zostać zdefiniowane w następujący sposób.

Wzdłuż warstwy płynu przepływu nieturbulentnego:

$$\text{Ciśnienie statyczne} + \text{ciśnienie dynamiczne} = \text{const}$$

Równanie Bernoulliego przyjmuje postać:

$$\frac{|\vec{V}|^2}{2} + gh + \frac{p}{\rho} = \text{constant}$$

Równanie 8 Równanie Bernoulliego

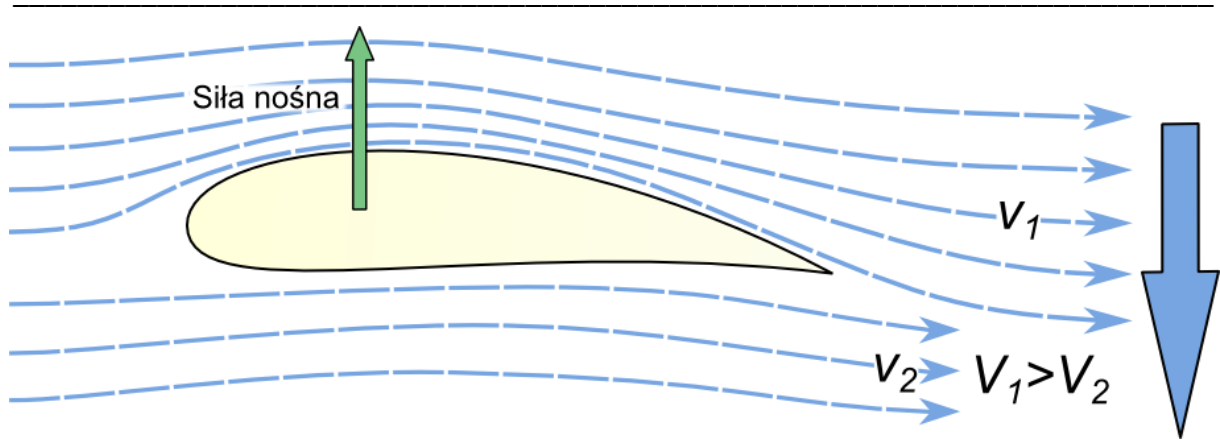
gdzie:

- $V$  - prędkość cieczy
- $g$  - przyspieszenie ziemskie
- $h$  - wysokość
- $p$  – ciśnienie
- $\rho$  – gęstość cieczy

Trzy człony równania określają odpowiednio:

- $\frac{|\vec{V}|^2}{2}$  – energia kinetyczna
- $gh$  – energia potencjalna
- $\frac{p}{\rho}$  – energia wewnętrzna (ciśnienia)

Z równania Bernoulliego możemy więc wywnioskować że im większa prędkość przepływu tym mniejsze ciśnienie płynu. Spostrzeżenie to jest wykorzystywane przy konstrukcji profili skrzydeł. W przeciwieństwie do płaskiego ciała położonego zgodnie z ruchem przepływających mas powietrza (Rys. 2.2 c), które w takiej sytuacji nie wytwarza siły nośnej skrzydło szybowca będzie zachowywać się zupełnie inaczej. Spowodowane jest to tym, że projekt profilu zmusza strugi powietrza opływające skrzydło od góry do zwiększenia prędkości o wiele bardziej niż strugi powietrza opływające od dołu, co powoduje różnice ciśnień nad i pod skrzydłem (Rys. 2.6). Taki wpływ kształtu na zachowanie się w strumieniu powietrza jak i inne cechy doprowadziły do badań nad zachowaniem różnych profili lotniczych. Powiemy na ten temat więcej w rozdziale 2.2.2.



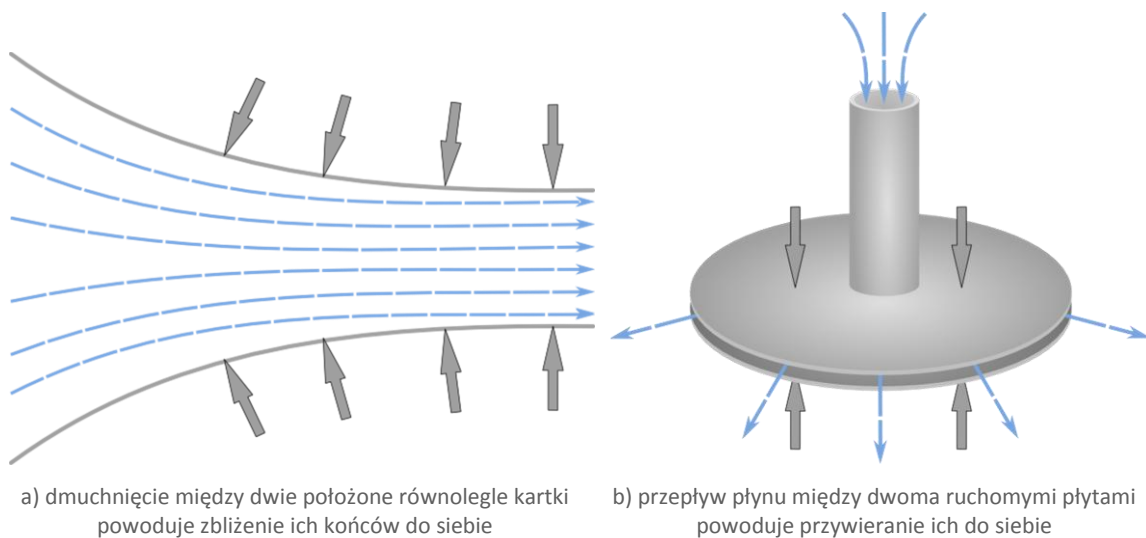
Dzięki odpowiedniemu wyprofilowaniu prędkość strug powietrza opływających profil od góry jest większa niż opływających od dołu ( $v_1 > v_2$ ). Z prawa Bernoulliego otrzymujemy różnicę ciśnień  $p_1 < p_2$ . Występowanie niższego ciśnienia nad profilem powoduje powstanie siły nośnej

Rys. 2.6 Przykładowy profil skrzydła w ruchu



### Paradoks hydrodynamiczny

Równanie Bernoulliego rozwiązuje problem tzw. Paradoksu hydrostatycznego. Mianem paradoksu hydrostatycznego określane jest spostrzeżenie, że ciśnienie płynu idealnego przepływającego przez przewężenie zmniejsza się, co wydaje się niezgodne ze zdrowym rozsądkiem. Błąd w potocznym rozumowaniu polega na przyjęciu, że płyn zmniejsza swoją objętość proporcjonalnie do objętości obszaru, w którym się znajduje. W rzeczywistości jednak ściśliwość w przypadku małych prędkości jest znikoma co powoduje, że płyn w przewężeniu nie zmniejsza swojej objętości, a zamiast tego rośnie jego prędkość, co zgodnie z równaniem Bernoulliego prowadzi do spadku ciśnienia.



a) dmuchnięcie między dwie położone równoległe kartki powoduje zbliżenie ich końców do siebie

b) przepływ płynu między dwoma ruchomymi płytami powoduje przywieranie ich do siebie

Rys. 2.7 Przykłady efektów wynikających z różnicy ciśnień płynu pozostającego w spoczynku i ruchomego, w których możemy zaobserwować tzw. paradoks hydrodynamiczny

## 2.1.2 Opór

W tym rozdziale zajmiemy się różnymi rodzajami oporu działającego na lecący szybowiec. Istnieją trzy rodzaje oporu:

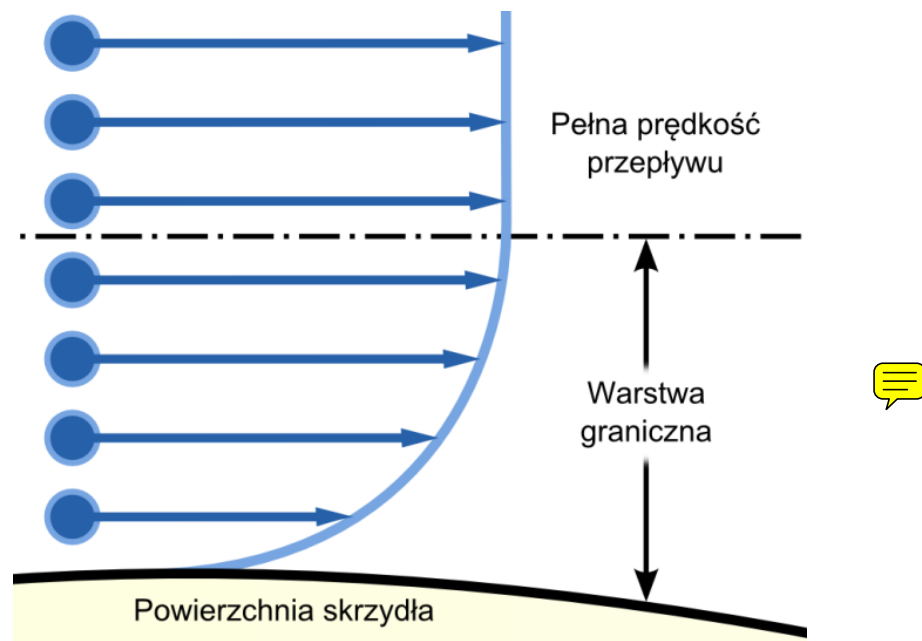
- Opór tarcia, wynikający z lepkości powietrza
- Opór profilu, wynikający z rozkładu ciśnienia na skrzydle (składowa siły aerodynamicznej równoległa do kierunku ruchu)
- Opór indukowany, wynikający ze skończonej rozpiętości skrzydeł

Można zastosować jeszcze dwie klasyfikacje grupujące powyższe opory:

- Ze względu na ilość rozpatrywanych wymiarów
  - Opór w 2D – do niego należą opór tarcia i profilu
  - Opór w 3D – indukowany
- Ze względu na pochodzenie oporu
  - Opór tarcia
  - Opór ciśnienia – tu profilu i indukowany

### 2.1.2.1 Opór tarcia

Opór tarcia, jak zostało wspomniane wyżej pochodzi od niezerowej lepkości cieczy  $\nu$  [kg/ms]. Dla powietrza w temperaturze 20°C i przy ciśnieniu 1 atm opór wyniesie  $\mu=18 \cdot 10^{-6}$  kg/ms. Dla porównania, przy tych samych warunkach dla wody  $\mu=1 \cdot 10^{-3}$  kg/ms.



Rys. 2.8 Rozkład prędkości strug powietrza w warstwie granicznej

Działanie lepkości cieczy jest istotne ze względu na występowanie warstwy granicznej tuż przy powierzchni szybowca. Bardzo cienka warstwa powietrza, zamiast opływać szybowiec, przemieszcza się wraz z nim. W wyniku tego w pełni wykształconym przepływie wzdłuż osi  $x$  występuje gradient  $x$ -owej składowej prędkości w kierunku osi  $y$ . Sytuacja ta została schematycznie przedstawiona na rysunku Rys. 2.8. Przed wprowadzeniem kolejnego pojęcia, wprowadzona zostanie jeszcze jedna bezwymiarowa wielkość – Liczba Reynoldsa określająca stosunek wielkości sił inercjalnych do sił lepkości (Równanie 9). Wraz z jej wzrostem coraz większy wpływ mają siły lepkości, które powyżej pewnej wielkości granicznej całkowicie odmieniają charakterystykę warstwy granicznej. Jak widać ze wzoru, wzrost  $R_e$  wywołać może wzrost prędkości przepływu, ale również wzrost długości opływającego ciała – a więc np. długości profilu skrzydła (ciężki). Sama liczba Reynoldsa stosowana jest w aerodynamice również jako tzw. liczba podobieństwa – gdy bada się przeskalowany (pomniejszony) element w tunelu aerodynamicznym, aby uzyskać dane odpowiadające rzeczywistości należy odpowiednio przeskalować (zwiększyć) prędkość przepływu tak, aby liczba Reynoldsa pozostała niezmienną.

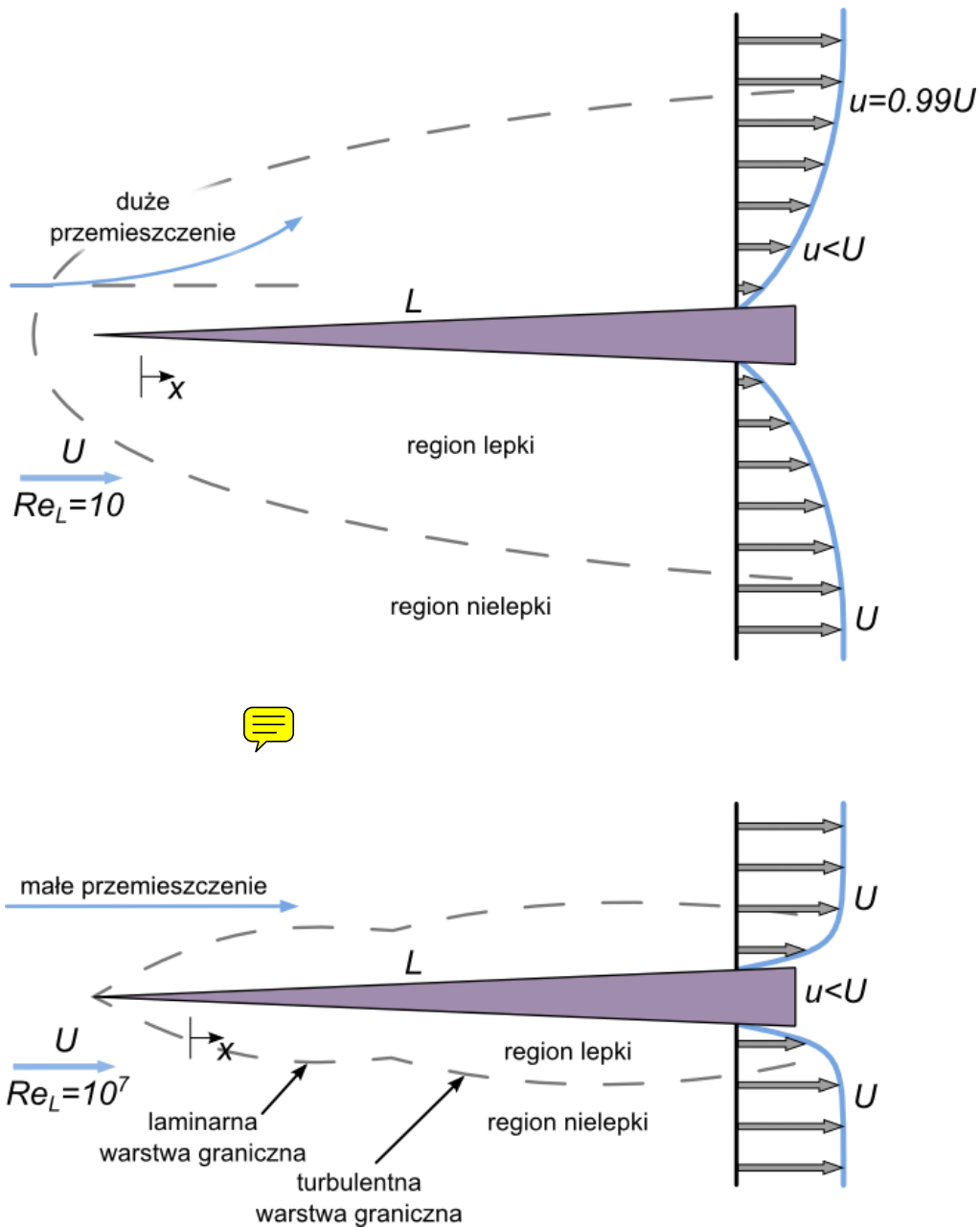
$$R_e = \frac{U_{ref} L_{ref}}{\nu}$$

Równanie 9 Liczba Reynoldsa

w powyższym równaniu:

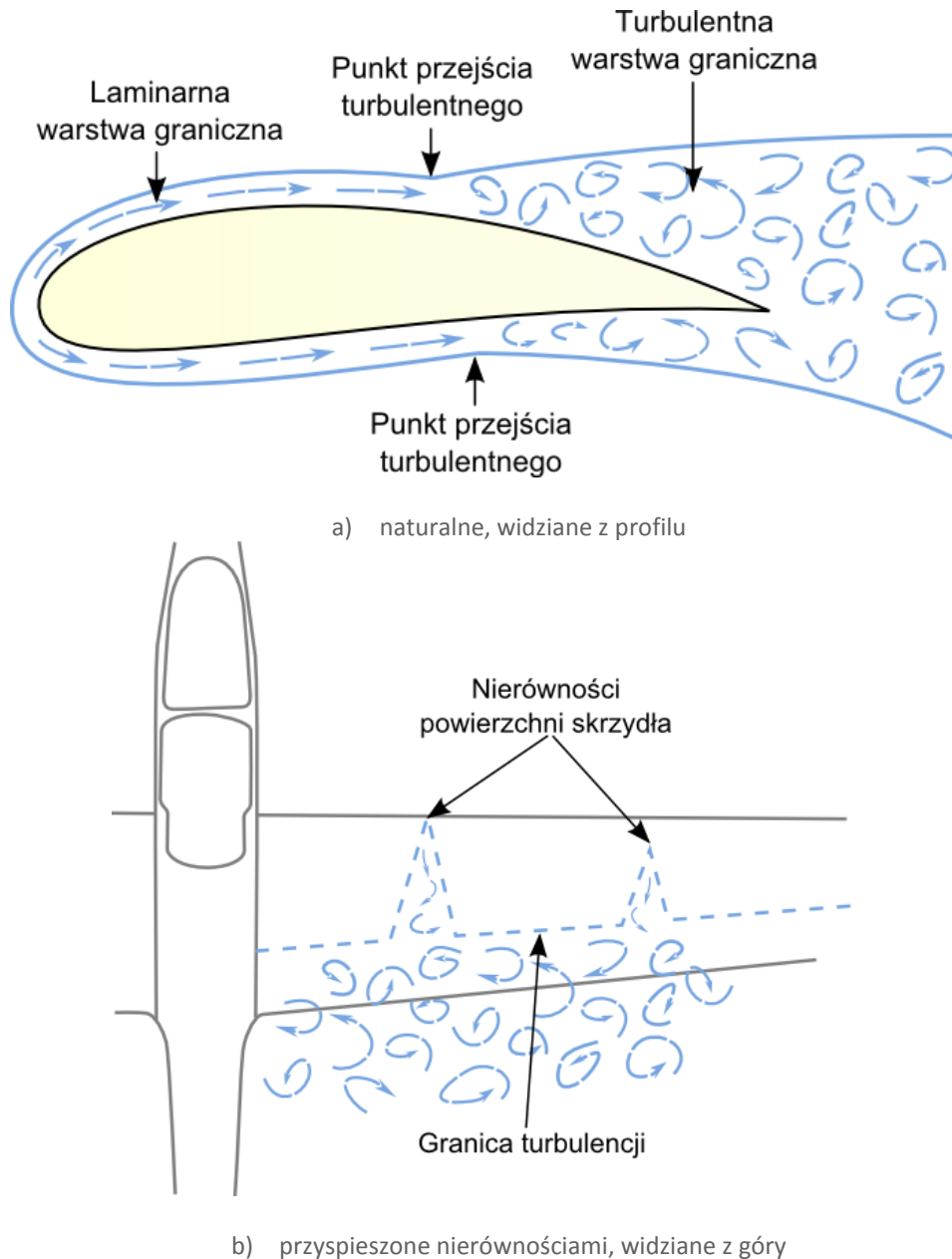
- $R_e$  – liczba Reynoldsa
- $U_{ref}$  – prędkość referencyjna. Pod tym pojęciem należy rozumieć uśrednioną prędkość odnoszącą się do całości zagadnienia
- $L_{ref}$  - wymiar charakterystyczny, czyli wymiar zjawiska o charakterze pewnej odległości mającej bezpośredni wpływ na stateczność ruchu płynu
- $\nu$  – lepkość

W przypadku szybowców należy również wziąć pod uwagę występowanie turbulentnej warstwy granicznej. Liczba Reynoldsa  $Re_x = 5 * 10^5$  umownie rozgranicza przepływ liniowy od turbulentnego. Różnice pomiędzy takimi przepływami pokazuje Rys. 2.9.



Rys. 2.9 Porównanie liniowej i turbulentnej warstwy granicznej.

Pojawienie się turbulentnej warstwy granicznej ma negatywny wpływ na wydajność skrzydła, gdyż powoduje wzrost oporu tarcia. Dlatego też zwykle próbuje się stworzyć skrzydła o jak największym udziale przepływu liniowego a jak najmniejszym turbulentnego. Na łatwość przemiany wpływa w bardzo dużym stopniu gradient ciśnienia wzdłuż osi przepływu – dodatni przyspiesza, a ujemny opóźnia przejście. Kolejnym czynnikiem ułatwiającym wystąpienie przepływu turbulentnego są wszelkiego rodzaju nierówności powierzchni (Rys. 2.10).



Rys. 2.10 Przejście opływu liniowego w turbulentny na skrzydle szybowca

Obliczenie wartości oporu pochodzącego od tarcia w ogólności nie jest sprawą prostą. O ile w przypadku w pełni liniowego przepływu nad płaską powierzchnią można sobie z tym poradzić za pomocą wyznaczonych analitycznie wzorów (Równanie 10), o tyle w przypadku przepływu turbulentnego należy zastosować sztucznie wprowadzone wzory, które mają dać wyniki zbliżone do empirycznych danych. Zwykle do uzyskania danych liczbowych stosuje się tutaj numeryczne symulacje. Sprawa jeszcze bardziej się komplikuje, gdy w przepływie występuje gradient ciśnienia. Na szczęście, znając charakterystykę profilu naszego skrzydła będziemy mogli założyć odległość miejsca przejścia od krawędzi natarcia i wyliczyć całkowity współczynnik oporu z empirycznego wzoru dla połączonego przepływu liniowego i turbulentnego (Równanie 11).

$$C_D = \frac{2D(L)}{\rho U^2 b L} = \frac{1.328}{\sqrt{Re_L}}$$

Równanie 10 Współczynnik oporu dla liniowej warstwy granicznej

$$C_D = 0,073 Re_L^{-0,2} + \left(\frac{x_{tr}}{L}\right)(1,328 Re_{tr}^{-0,5} - 0,073 Re_{tr}^{-0,2})$$

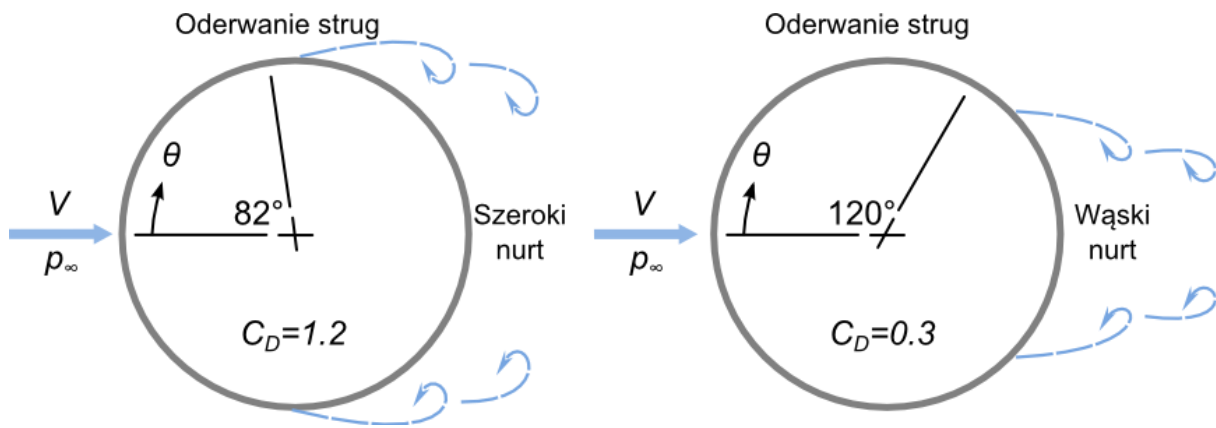
Równanie 11 Współczynnik oporu dla warstwy granicznej liniowej od 0 do  $x_{tr}$  a turbulentnej od  $x_{tr}$  do L.

Gdzie:

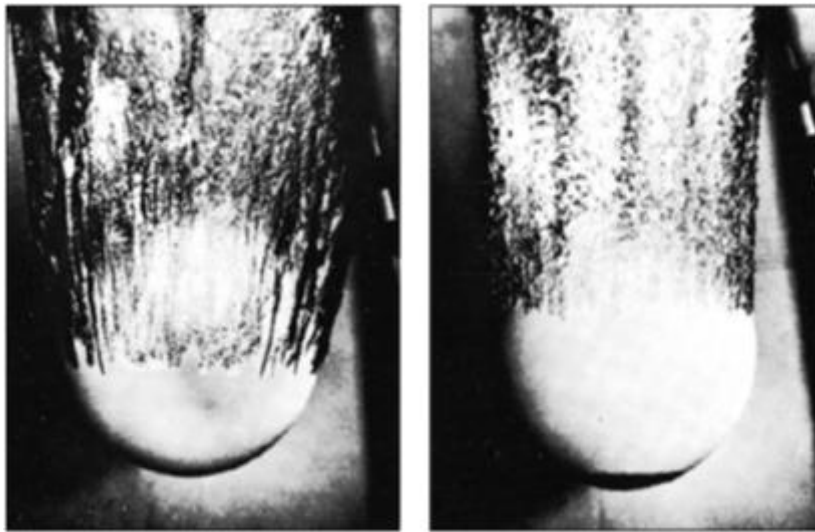
- D – wartość siły oporu
- L – długość powierzchni
- b – szerokość powierzchni
- $x_{tr}$  – odległość punktu przejściowym od krawędzi natarcia (przemiany z warstwy liniowej na turbulentną)
- $Re_L$  – wartość liczby Reynoldsa na końcu powierzchni L
- $Re_{tr}$  – wartość liczby Reynoldsa w punkcie przejściowym  $x_{tr}$
- U – prędkość niezabuzonego przepływu,  $\rho$  – gęstość cieczy

Pojawienie się turbulentnej warstwy granicznej jest zwykle niekorzystne ze względu na zwiększenie oporu powodowane rozszerzeniem warstwy granicznej. Mimo to, w przypadku tzw. oporu kształtu sytuacja jest odwrotna. Dotyczy on opływu ciał o dużym przekroju poprzecznym w porównaniu do ich długości. Takiej sytuacji towarzyszy zawsze oderwanie strug – wtedy to linie prądu przestają przypominać kształtem opływany przedmiot, a w obszarze nurtu przepływ jest chaotyczny. W takiej sytuacji pomocne okazuje się wymuszenie powstania turbulentnej warstwy granicznej, która ogranicza zakres powstawania oderwania i zmniejsza nurt. Rys. 2.11 pokazuje wpadającą do wody kulę do kręgli oraz kulę do kręgli z wprowadzonymi na powierzchnię nierównościami wielkości ziarenek piasku. Widać wyraźnie, że turbulentna warstwa graniczna wpłynęła na zmniejszenie wielkości nurtu. Tego typu wymuszanie przejścia spotykane jest w przypadku faktury piłki golfowej, czy włosków na piłce tenisowej. W końcu w niektórych samolotach również wymusza się to przejście, aby zapobiec oderwaniu strug od skrzydeł – tzw. Przeciągnięcie – zjawisko szczególnie niebezpieczne i uniemożliwiające jakąkolwiek kontrolę nad lotem.





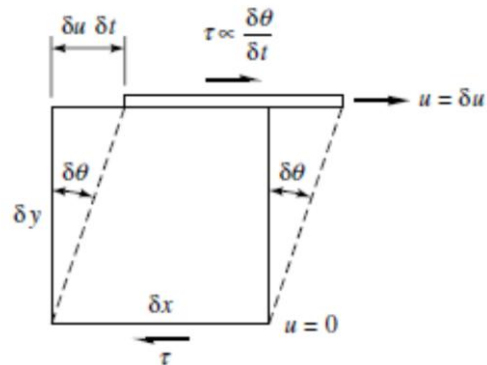
a) Schematyczny rysunek. Po lewej liniowa, a po prawej turbulentna warstwa graniczna



b) Przykład – kula do kręgli wpadająca do wody. Po lewej liniowa, a po prawej turbulentna warstwa graniczna

Rys. 2.11 porównanie kąta oderwania i rozmiaru nurtu

### Wyprowadzenie współczynnika lepkości



Rys. 2.12 Wyprowadzenie współczynnika lepkości

Jak widać na **Błąd! Nie można odnaleźć źródła odwołania.** przyłożenie napięcia ścinającego  $\tau$  spowoduje przesunięcie warstwy cieczy, proporcjonalne do zmiany kąta w jednostce czasu.

$$\tau \propto \frac{\delta\theta}{\delta t}$$

Korzystając następnie z trygonometrycznych własności zapisujemy

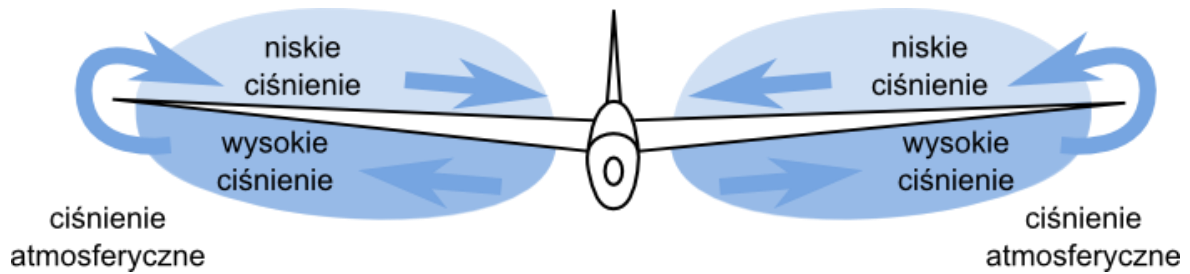
$$\tan \delta\theta = \frac{\delta u \delta t}{\delta y}$$

W końcu przybliżając dla małego kąta ( $\tan \delta\theta = \delta\theta$ ) otrzymujemy

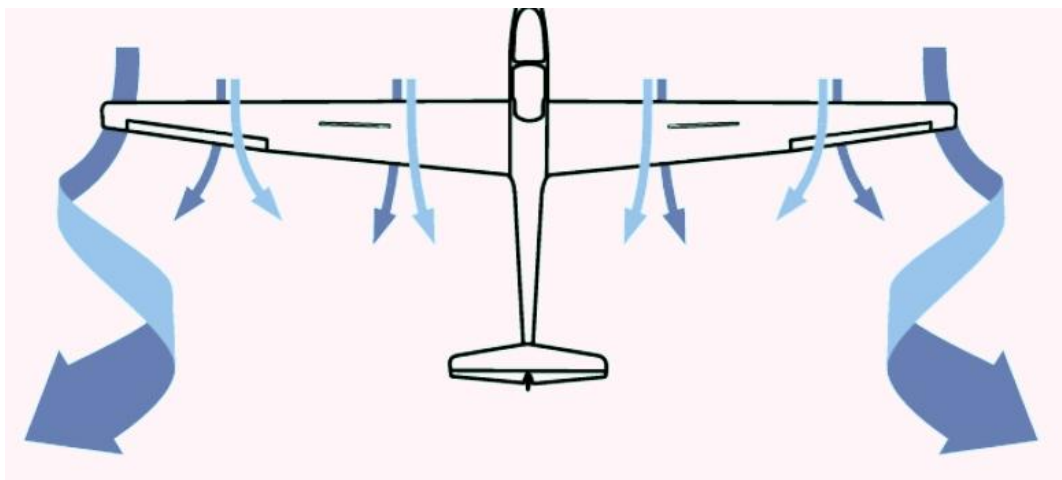
$$\frac{d\theta}{dt} = \frac{du}{dy}, \text{ a więc możemy zapisać } \tau = \mu \frac{d\theta}{dt} = \mu \frac{du}{dy}$$

Rysunek i wyprowadzenie za „Fluid Mechanics” – F.M. White.

### 2.1.2.2 *Opór indukowany*



a) Schematyczne przedstawienie ciśnienia na płatach szybowca



b) Schematyczny widok odchylonych linii prądu i wirów końcówek (rzut z góry)



c) Zdjęcie wiru powstałego za samolotem pasażerskim przelatującym przez powietrze zabarwione świecą dymną

Rys. 2.13 Wiry powstające na końcówkach skrzydeł

Powstanie oporu indukowanego bierze się z efektów występujących na końcówkach skrzydeł. Występowanie dodatniego współczynnika siły nośnej wymaga występowania różnicy ciśnień między dolną o górną powierzchnią skrzydła. Powietrze z dołu, pod (po uśrednieniu) wyższym ciśnieniem stara się przedostać na górę, gdzie (średnio) panuje niższe ciśnienie – dzieje się to właśnie na końcówkach skrzydeł. Taki ruch powietrza owocuje powstaniem wirów na końcówkach. Do wytworzenia wirów i odchylenia strumieni cieczy potrzebna jest odpowiednia energia, która bierze się z pojawienia się dodatkowego oporu – indukowanego. To zjawisko niejako skraca rozpiętość skrzydeł – na końcach ciśnienie górnej i dolnej części płata wyrównuje się, więc nie działają już one jako powierzchnie nośne. Przesunięcie górnych linii prądu ku osi symetrii samolotu a dolnych ku końcówkom skrzydeł powoduje powstanie wirów na całej rozpiętości skrzydła. Znajomość rozkładu wirów oraz kilka przekształceń matematycznych pozwala uzyskać wyrażenia opisujące współczynnik indukowanego oporu oraz współczynnik indukowanej siły nośnej.

$$C_L = \frac{1}{\frac{1}{C_l'} + \frac{1}{\pi\Lambda}} (\alpha + \beta)$$

Równanie 12 współczynnik siły nośnej dla skrzydła o skończonej długości

gdzie  $\alpha$  to kąt natarcia a  $\beta$  oraz  $C_l'$  to parametry profilu skrzydła.

$$C_{Di} = \frac{C_L^2}{\pi\Lambda} (1 + \delta)$$

Równanie 13 Współczynnik oporu indukowanego dla skrzydła o skończonej długości

gdzie  $\delta$  zależy od kształtu skrzydła.

W powyższych wzorach współczynnik siły nośnej opisuje całkowitą siłę nośną, a więc złożenie siły nośnej od profilu i od płata o skończonej rozpiętości. Opór odpowiada jedynie za część indukowaną. Podstawową charakterystyką skrzydła mającą wpływ na skalę zjawiska sił indukowanych jest wydłużenie  $\Lambda$ .

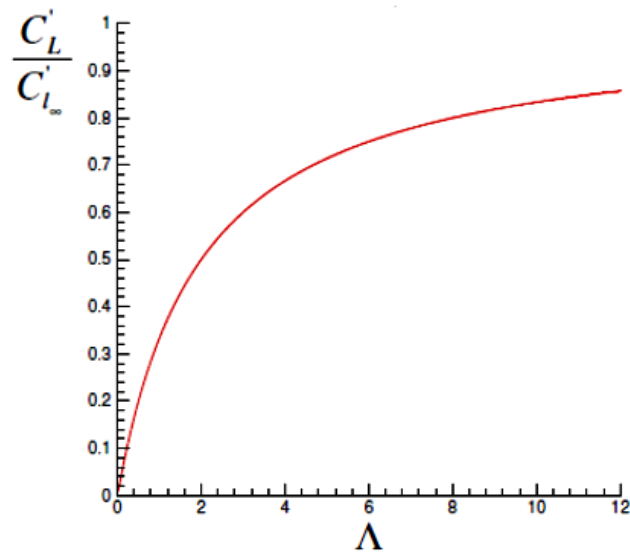
$$\Lambda = \frac{b^2}{c}$$

Równanie 14 Wydłużenie skrzydła

gdzie  $b$  to rozpiętość skrzydeł,  $c$  to cięciwa.

Im wydłużenie większe, tym wpływ sił indukowanych jest mniejszy. Oczywiście sama siła zjawiska nie zmienia się w zależności od rozpiętości, za to wydłużenie skrzydła powoduje

wzrost siły nośnej w porównaniu, do której ubytek na końcówkach względnie jest mniejszy. Aby to zobrazować, na poniższej ilustracji (Rys. 2.14) pokazemy wykres stosunku pochodnej po kącie natarcia współczynnika siły nośnej profilu rzeczywistego do profilu o nieskończonej rozpiętości, dla założenia, że parametr  $C'_l = 2\pi$ . W takiej sytuacji porównanie pochodnych przyjmuje postać  $\frac{C'_L}{C'_{L_\infty}} = \frac{\Lambda}{\Lambda+2}$



Rys. 2.14 Wpływ zwiększenia wydłużenia skrzydła na istotność poprawki na indukowaną siłę nośną

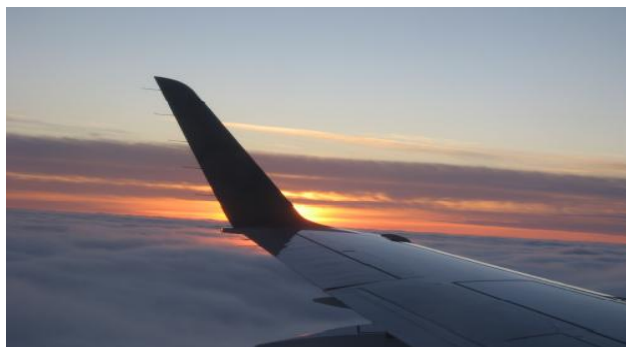
### 2.1.2.3 Rozwiązania problemów ze skończoną rozpiętością skrzydeł

Najprostszym i stosunkowo skutecznym sposobem na obniżenie wpływu indukowania się oporu i siły nośnej na końcówkach skrzydła jest zastosowanie wingletów. Są to panele prostopadłe do powierzchni nośnej, które mają na celu uniemożliwienie przemieszczania się powietrza pomiędzy górną a dolną płaszczyzną. Szczególnie istotne jest to dla krótkich skrzydeł, czego przykład można zobaczyć na przykładzie tylnego spoilera auta F1.



Rys. 2.15 Tylni spoiler auta F1 zakończony wingletami

W niektórych samolotach pasażerskich dysponujących dużą rozpiętością skrzydeł mimo wszystko stosuje się winglety aby dodatkowo poprawić efektywność skrzydła, ale również aby ograniczyć rozmiary wirów powstających za startującymi i lądującymi samolotami. Dzięki temu mogą one korzystać z pasów startowych jedne po drugich w mniejszych odstępach czasu.



Rys. 2.16 Skrzydło samolotu zakończone wingletami

Kolejnym sposobem na obniżenie indukowanego oporu jest zastosowanie odpowiedniego kształtu skrzydła. Analitycznie można wyliczyć iż najefektywniejszym pod tym względem kształtem jest elipsa, co zrealizowano jak tylko technologia umożliwiła skonstruowanie wystarczająco wytrzymałego skrzydła w tym kształcie. Pierwszym i ostatnim samolotem o takich skrzydłach był Supermarine Spitfire. Mimo sukcesu, nie powielono tej konstrukcji, gdyż trapezoidalne zwężanie skrzydeł ku końcówkom daje porównywalny efekt przy ogromnym uproszczeniu procesu produkcji.



Rys. 2.17 Supermarine Spitfire ze skrzydłami w kształcie elipsy

Ostatnim przykładem może być próba stworzenia skrzydła o nieskończonej rozpiętości – dwupłatowiec, którego górny i dolny płat stykały się na końcach tworząc pierścień dookoła kadłuba. Taki samolot powstał i był zdolny do lotu, jednak sterowanie nim przysparzało wielu trudności.



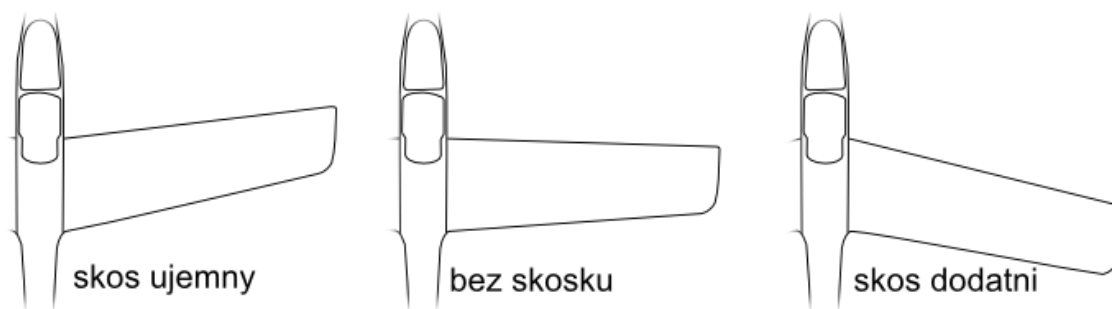
Rys. 2.18 Samolot ze skrzydłami w kształcie pierścienia dookoła kadłuba

## 2.2 Konstrukcja szybowca

Możemy wyróżnić wiele rodzajów układów konstrukcyjnych szybowców. Pozwala to klasyfikować szybowce na wiele sposobów na przykład poprzez rozróżnienie usytuowania skrzydeł względem kadłuba:

- Struktura górnopłata – odznacza się tym, że skrzydła umieszczone są nad kadłubem lub w jego górnej części. Układ ten był bardzo powszechny w początkach szybownictwa następnie ustępując miejsca układowi średniopłata
- Struktura dolnopłata – w tym układzie skrzydła umieszczane są w dolnej części kadłuba. Układ ten jest w szybownictwie praktycznie niestosowany z powodu niskiej stateczności oraz małej odległości skrzydeł od ziemi w trakcie lądowania, co zwiększa możliwość uszkodzenia szybowca
- Struktura średniopłata – układ, w którym skrzydła znajdują się mniej więcej w połowie wysokości kadłuba, jest zarazem najczęściej występującym układem

Możemy także rozróżniać szybowce na podstawie różnic konstrukcji usterzeń o czym powiemy sobie w rozdziale 0. Innym czynnikiem klasyfikacji szybowców może być rodzaj zastosowanych skosów skrzydeł.



Rys. 2.19 Rodzaje skosów skrzydeł

Jednym z podstawowych elementów szybowca jest jego kadłub, który od strony wydajności lotu jest elementem szkodliwym, ponieważ nie produkuje siły nośnej, a jedynie wpływa na wzrost oporu. Jest on jednak niezbędny, jako połączenie płata z usterzeniami, mieszczący załogę z wyposażeniem i tworzący „pojazdową” część szybowca poruszającego się po ziemi. Ze względów optymalizacji wydajności kadłuby szybowców są więc kształtami bardzo opływowymi, przy konstrukcji których stawia się na minimalizację oporu poprzez stworzenie jak najbardziej opływowej i szczelnej sylwetki generującej minimalną ilość powstających w trakcie lotu turbulencji. Najważniejszymi elementami szybowca są skrzydła. Jako elementy generujące główną część siły nośnej są one najbardziej interesujące z perspektywy fizyki lotu i dlatego ich konstrukcji poświęcimy najwięcej uwagi w tym rozdziale. Omówimy także kolejny zestaw elementów, bez których szybowiec nie byłby zdolny do lotu - usterzenie.

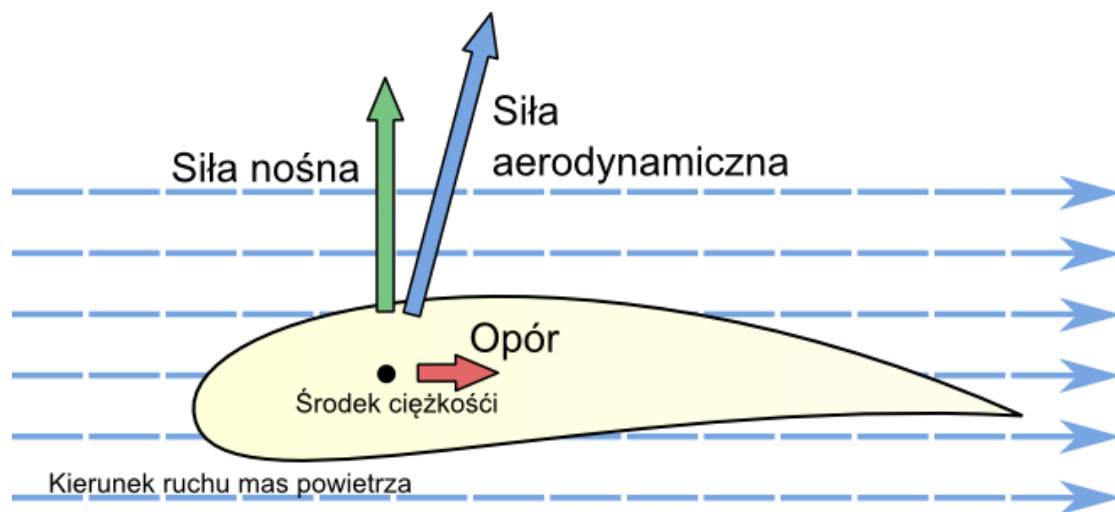


## 2.2.1 Powierzchnie nośne

W tym rozdziale zgłębimy pojęcie powierzchni nośnych, których doskonałym przykładem są skrzydła szybowca. Problemy aerodynamiki powierzchni nośnych zwykle dzieli się na dwie części:

- Przepływ dwuwymiarowy wokół profilu
- Efekty występujące na końcówkach skrzydeł

Powierzchnia nośna to, z definicji, powierzchnia która poruszając się w płynie generuje siłę nośną dużo większą od generowanego oporu (Rys. 2.20).



Rys. 2.20 Przykład powierzchni nośnej – skrzydło szybowca

Wymienione właściwości powierzchni nośnych są powodem ich szerokiego zastosowania w rozmaitych dziedzinach inżynierii, przy konstrukcji samolotów, statków, samochodów czy elektrowni wiatrowych (Rys. 2.21 a-d). Przykłady powierzchni nośnych możemy także odnaleźć w przyrodzie (Rys. 2.21 e-f).



a) samolot typu latające skrzydło



b) bolid F1



c) śruba okrętowa



d) łopatką wirnika turbiny wiatrowej



e) skrzydło orła przedniego

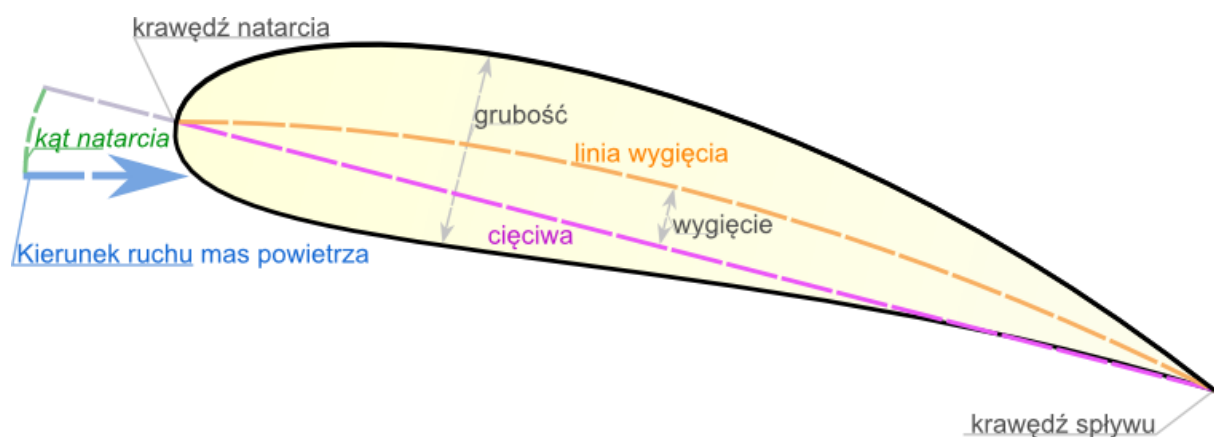


f) płetwa ogonowa humbaka

Rys. 2.21 Przykłady powierzchni nośnych

Przy opisywaniu profili skrzydeł stosuje się następujące nazewnictwo (Rys. 2.22):

- krawędź natarcia (nosek) – najbardziej wysunięta do przodu część profilu
- krawędź spływu (ostrze) – najbardziej wysunięta do tyłu część profilu
- cięciwa – linia łącząca krawędź natarcia z krawędzią spływu
- linia wygięcia – linia poprowadzona w połowie między górną i dolną krawędzią profilu
- grubość profilu – odległość między górną i dolną krawędzią profilu mierzona w jego najgrubszym punkcie prostopadłe do cięciwy
- wygięcie (strzałka ugięcia) – maksymalna odległość pomiędzy cięciwą a linią szkieletową.

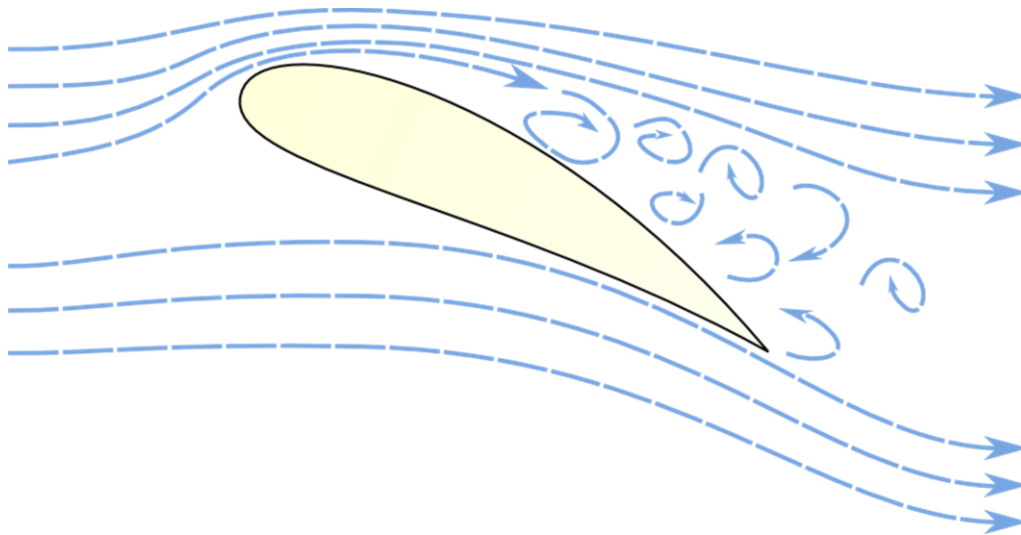


Rys. 2.22 Nomenklatura profilu lotniczego

Bardzo ważnym pojęciem przy opisie profili jest kąt natarcia. Jest to określenie kąta między prostą, na której leży cięciwa skrzydła, a kierunkiem opływających skrzydło mas powietrza. Kąt natarcia jest ważny, ponieważ jego wartość w ogromnym stopniu wpływa na wartość powstających sił aerodynamicznych. Im większy kąt natarcia tym większa jest wytwarzana siła aerodynamiczna, a co za tym idzie zarówno siła nośna jak i opór. Niestety przyrost siły nośnej przy zwiększającym się kącie natarcia ma swoje granice. Granicą tą jest tzw. kąt krytyczny. Dla kąta tego strumienie powietrza przestają opływać skrzydło w sposób laminarny i następuje powstawanie za skrzydłem silnych turbulencji. Wraz z dalszym przyrostem kąta natarcia powoduje to drastyczny spadek siły nośnej. Zjawisko to nazywamy przeciągnięciem (patrz Przeciągnięcie). Skrzydła posiadają konstrukcję niepełską. Są lekko wygięte do góry, co określamy przy pomocy linii wygięcia. Ma to korzystny wpływ na powstającą siłę nośną, ponieważ powoduje zwiększenie różnicy ciśnień pomiędzy górną i dolną warstwą skrzydła. Powoduje to, że skrzydło ustawione pod kątem natarcia  $0^\circ$  będzie wytwarzać siłę nośną, natomiast kąt, przy którym wytwarzana przez skrzydło siła nośna wyniesie 0 będzie lekko ujemny (Rys. 2.24).

### Przeciągnięcie

Jest to zjawisko nagłego spadku wartości siły nośnej i wzrostu wartości siły oporu. Spowodowane jest przekroczeniem krytycznego kąta natarcia lub zbyt małą prędkością lotu.



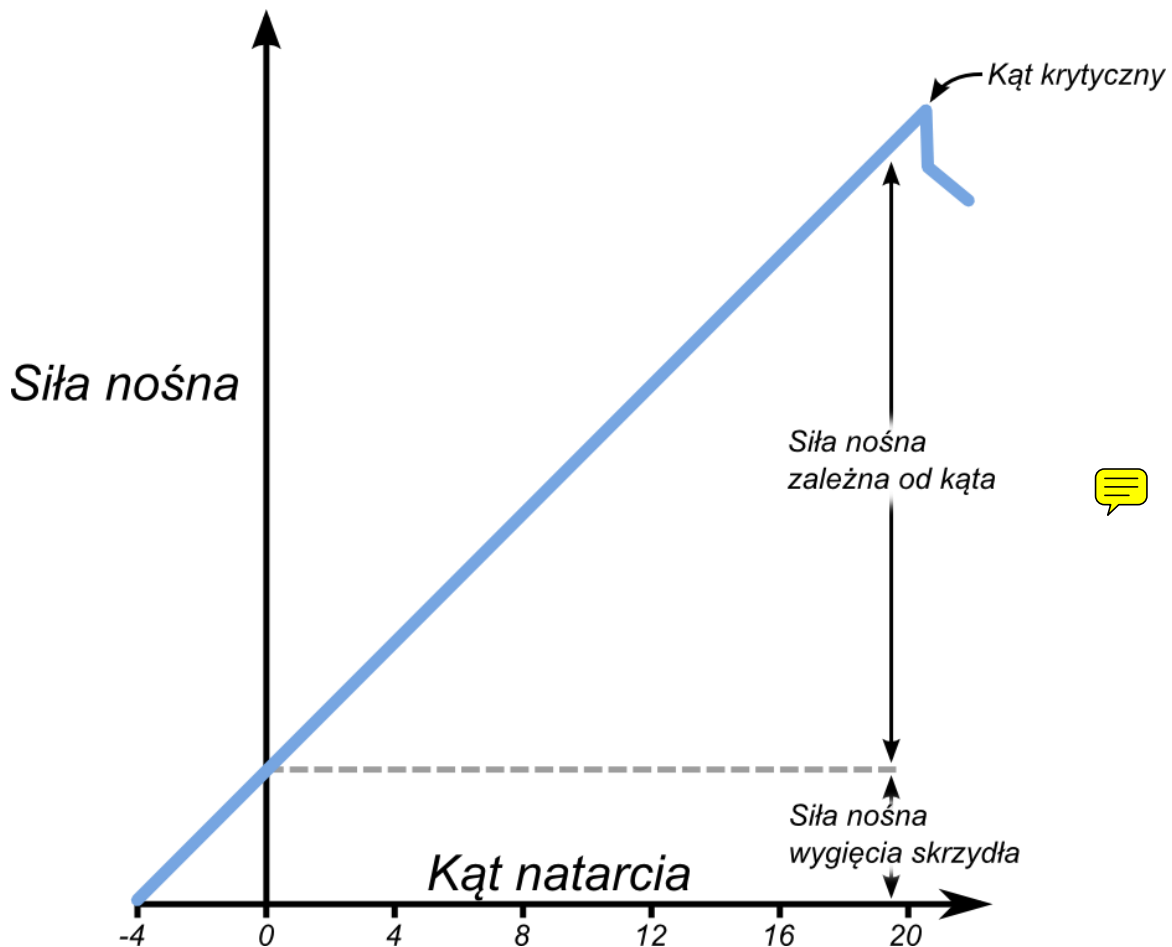
Rys. 2.23 Efekt przeciągnięcia

W zakresie "małych" kątów natarcia wzrost kąta powoduje wzrost siły nośnej, a więc ruch samolotu do góry - czyli w efekcie kąt natarcia maleje. To negatywne sprzężenie zwrotne zapewnia stabilny lot. Po przekroczeniu kąta krytycznego, mechanizm ten działa odwrotnie - sprzężenie staje się dodatnie. Wiąże się to z utratą sterowności. Moment przeciągnięcia związany jest z oderwaniem się linii prądu od powierzchni płata.

Można wyróżnić dwa rodzaje przeciągnięcia.

- Przeciągnięcie statyczne - spowodowane przez zbyt małą prędkość samolotu. Kiedy lot odbywa się po linii prostej, ze zmniejszającą się prędkością (np. podczas podchodzenia do lądowania), aby utrzymać stałą wartość sił nośnej należy zwiększać kąt natarcia. Stąd też przy charakteryzowaniu własności szybowca czy innego statku powietrznego mówi się o prędkości, a nie o kącie przeciągnięcia.
- Przeciągnięcie dynamiczne - powstałe na skutek gwałtownej zmiany kąta natarcia. Wtedy szybowiec "nie zdąży" podążać za wzrostem sił nośnej i efekt sprzężenia zwrotnego zadziała zbyt późno. Takie przeciągnięcie może wystąpić przy prędkościach znacznie większych od prędkości przeciągnięcia, np. w wyniku wykonania bardzo ostrego skrętu. Wystąpieniu takiego przeciągnięcia sprzyja występowanie turbulencji, a ogranicza jego prawdopodobieństwo zachowanie zapasu prędkości podczas wykonywania manewru skrętu.

Aby powrócić do normalnego lotu i odzyskać kontrolę nad maszyną, należy zmniejszyć kąt natarcia - skierować drążek sterowania do przodu i poczekać aż samolot znów zacznie prawidłowo reagować na stery. Głównym problemem - poza związanymi z oderwaniem linii prądu wibracjami - jest spadek wysokości jaki następuje podczas przeciągnięcia i wyprowadzania z niego. Dlatego też, jeśli dochodzi do przeciągnięcia na dużej wysokości, jest to o wiele mniej groźne niż podczas startu lub lądowania, gdzie brakuje wysokości na przeprowadzenie niezbędnych manewrów.

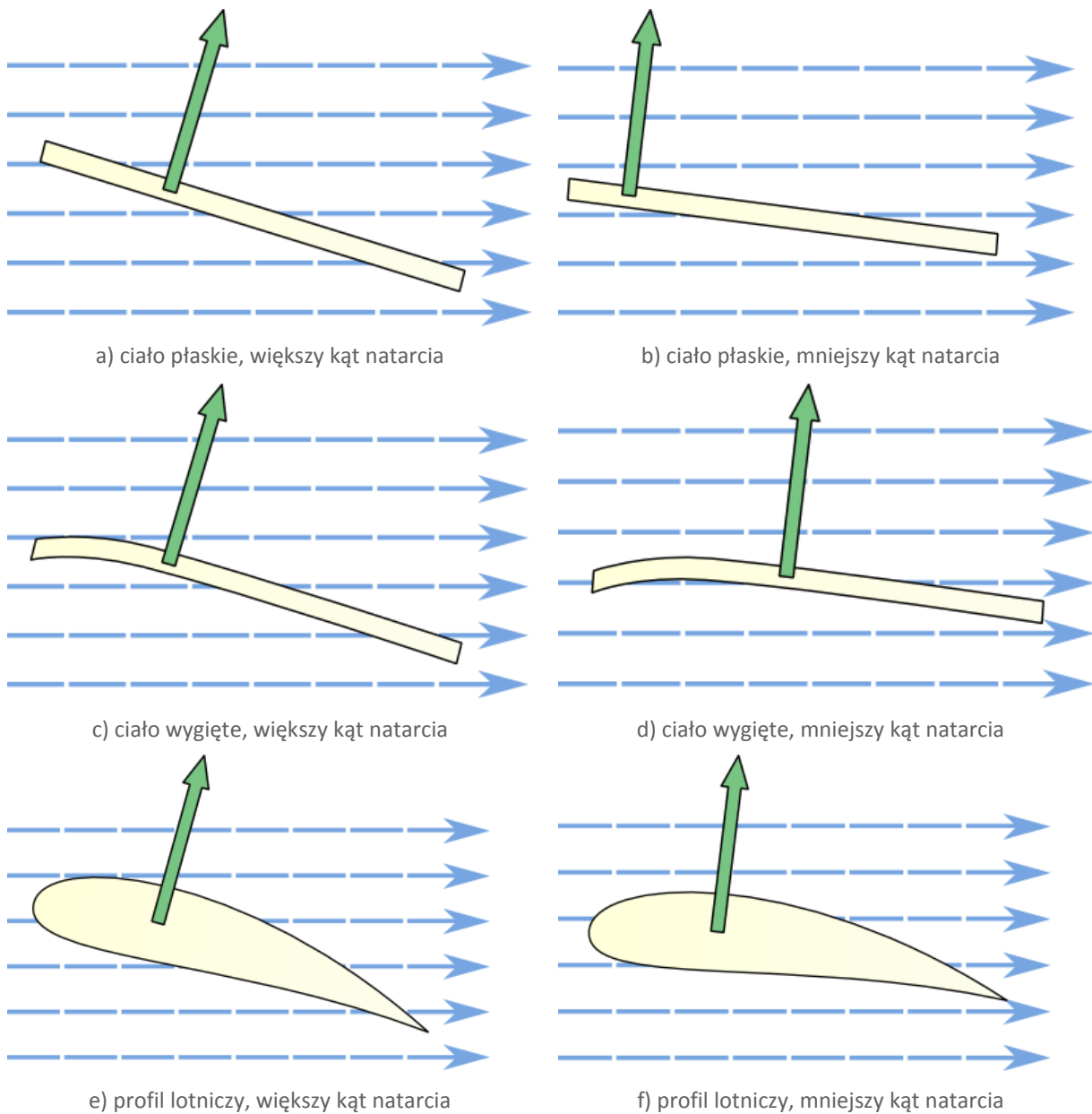


Rys. 2.24 Siła nośna jako funkcja kąta natarcia

Krzywizna skrzydła powoduje lekkie zwiększenie kąta krytycznego. Wczesne eksperymenty w początkach szybownictwa wykazały, że lekka krzywizna może spowodować zmianę kąta krytycznego z 10 stopni dla płaskiego skrzydła, nawet aż do 16 stopni, co oznaczało, że może zostać osiągnięta o wiele większa siła nośna, a także zmniejszało szanse narażenia lotnika na omyłkowe przekroczenie dopuszczalnych kątów nachylenia. Następnym ważnym parametrem zmieniającym się wraz z lekkim wykrzywieniem skrzydła szybowca jest przesunięcie środka parcia skrzydła poruszającego się w powietrzu.

Środek parcia jest to punkt do którego przyłożona jest wypadkowa sił aerodynamicznych i aerostatycznych. Wyznaczenie tego punktu jest o tyle ważne, że w tym miejscu nie występuje moment sił i korzystne jest umiejscowienie w nim (lub dookoła niego) elementów konstrukcyjnych odpowiedzialnych za wytrzymałość skrzydła. Dla płaskiego skrzydła postawionego w poprzek do kierunku ruchu powietrza środek parcia znajduje się dokładnie w środku skrzydła. Wraz z redukcją kąta nachylenia środek parcia przesuwa się do przodu (Rys. 2.25a,b). Dla skrzydła wygiętego jest jednak inaczej. Dla kątów poniżej kąta krytycznego możemy zaobserwować (w zależności od rozmiaru wygięcia) sytuację, w której wraz z redukcją kąta nachylenia środek parcia przesuwa się do tyłu skrzydła (Rys. 2.25c,d). Spostrzeżenie to jest bardzo ważne, ponieważ ze względów wytrzymałościowych oraz

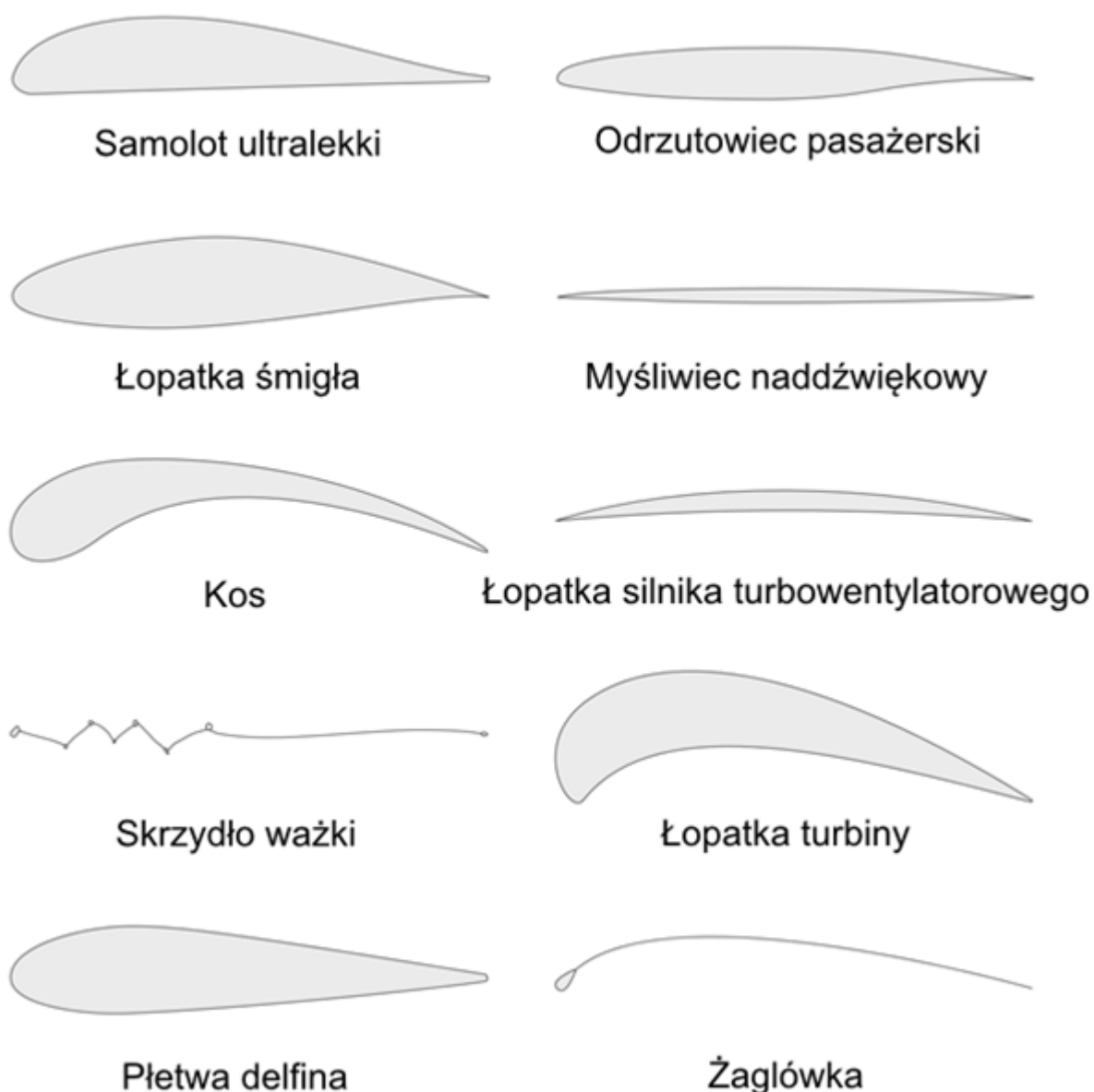
komfortu sterowania korzystne jest aby środek parcia skrzydła był jak najbardziej zbliżony do środka ciężkości szybowca. Zagadnienie to było bardzo dokładnie badane w czasach po drugiej wojnie światowej przy wykorzystaniu tuneli aerodynamicznych, gdzie poszukiwano krzywizny, która będzie się charakteryzować utrzymywaniem środka parcia blisko środka ciężkości dla szerokiej gamy kątów natarcia (Rys. 2.25e,f). Profile tego typu charakteryzują się wygięciem w okolicach 2-4% i są stosowane w większości nowoczesnych szybowców oraz samolotów.



Rys. 2.25 Środki parcia prostych powierzchni nośnych w zależności od kąta natarcia

## 2.2.2 Porównanie właściwości profili skrzydeł

W tym rozdziale przedstawimy klasyfikację profili skrzydeł oraz standardy dotyczące opisywania ich. Pokażemy popularne kształty, wykresy oporu i siły nośnej. Tutaj również pokażemy kształt i charakterystykę profilu Wortmann FX-73-K-170/22 użytego przy budowie szybowca DG-500, który modelowaliśmy.



Rys. 2.26 Przykłady profili powierzchni nośnych

Na samym początku, na rysunku Rys. 2.26 zaprezentujemy bez wdawania się w szczegóły cały szereg rozmaitych profili, aby pokazać różnorodność ich kształtu ze względu

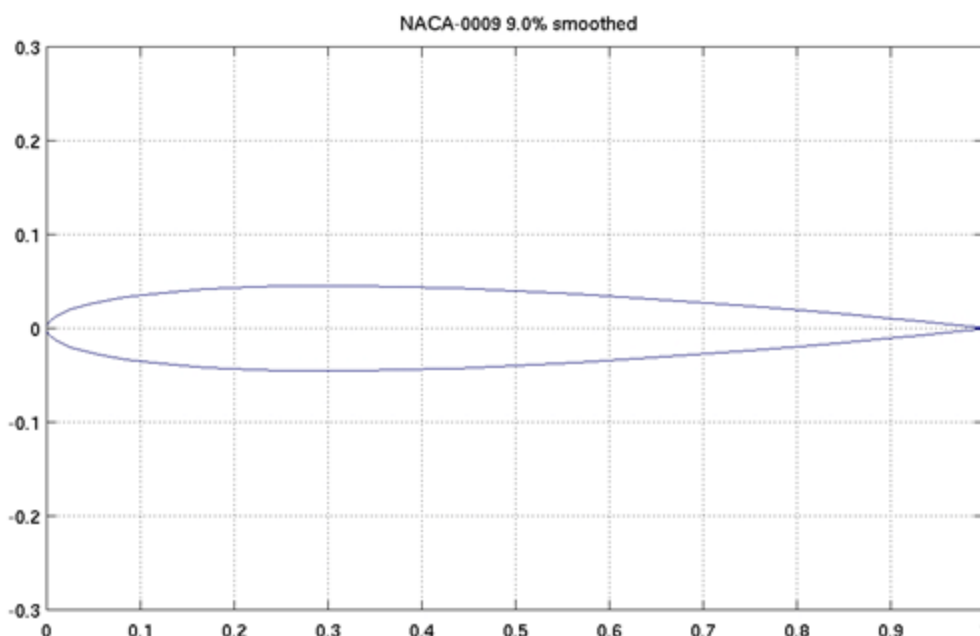
na spełnianie zadania. I tak po kolei mamy skrzydło ultralekkiego samolotu, skrzydło pasażerskiego odrzutowca, łopatkę śmigła i skrzydło ponadźwiękowego myśliwca. Już tutaj można zauważyć, że wraz ze wzrostem prędkości profile wysmuklają się. Większa prędkość zapewnia odpowiednią siłę nośną mimo spadku wartości współczynnika siły nośnej a zmniejsza się tym sposobem opór. Punkt maksymalnej grubości przesuwa się ku tyłowi, aby zminimalizować obszar przepływu turbulentnego. Profil myśliwca jest niemal symetryczny, aby nie tracił on wydajności przy odwróconym locie. Następnie znajduje się profil skrzydła kosa, łopatki wentylatora z silnika turbowentylatorowego, skrzydło ważki, łopatkę turbiny, płetwa delfina i w końcu żagiel łódki.

Pierwszą organizacją, która usystematyzowała sposób konstruowania i opisywania profili była amerykańska NACA – *National Advisory Committee for Aeronautics*. Działała ona od 1915 do 1958 roku, kiedy to została rozwiązana a w jej miejsce powołano *National Aeronautics and Space Administration* (NASA). NACA opracowała oznaczenia cztero-, pięcio- i sześć- cyfrowe.

Profile czterocyfrowe: NACA ABCD, gdzie:

- A – względne ugięcie w procentach  $f/c$
- B – umiejscowienie strzałki ugięcia, gdzie jeśli odległość strzałki od krawędzi natarcia to  $x_M$ , to  $B = 10x_M/c$
- CD – relatywna grubość profilu podana w procentach

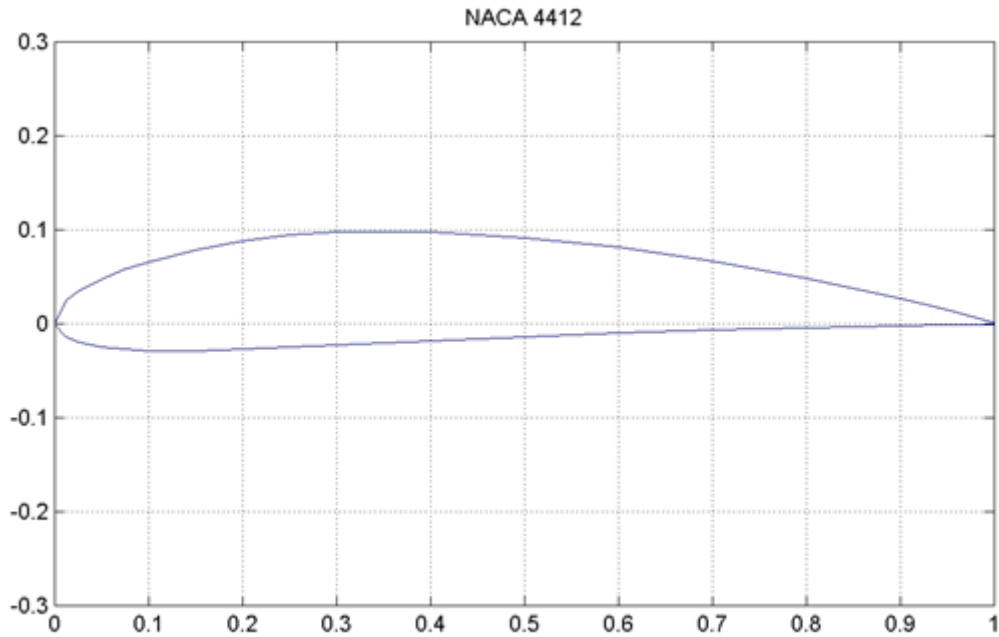
Poniżej, na znajdują się dwa przykładowe czterocyfrowe profile NACA.



Rys. 2.27 Profil NACA 0009

Na Rys. 2.27 widać profil symetryczny, pozbawiony ugięcia, o relatywnej grubości 9%.





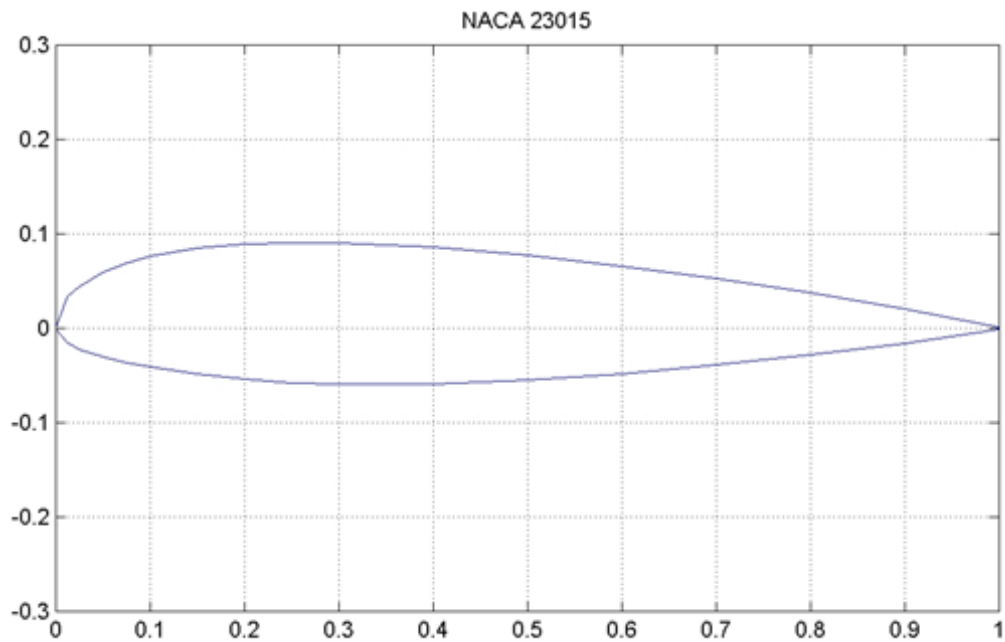
Rys. 2.28 Profil NACA 4412

Na rysunku Rys. 2.28 Profil NACA 4412 widać profil o względnym ugięciu 4%, strzałka ugięcia znajduje się w 0,4 cięciwy a grubość profilu to 12% długości cięciwy.

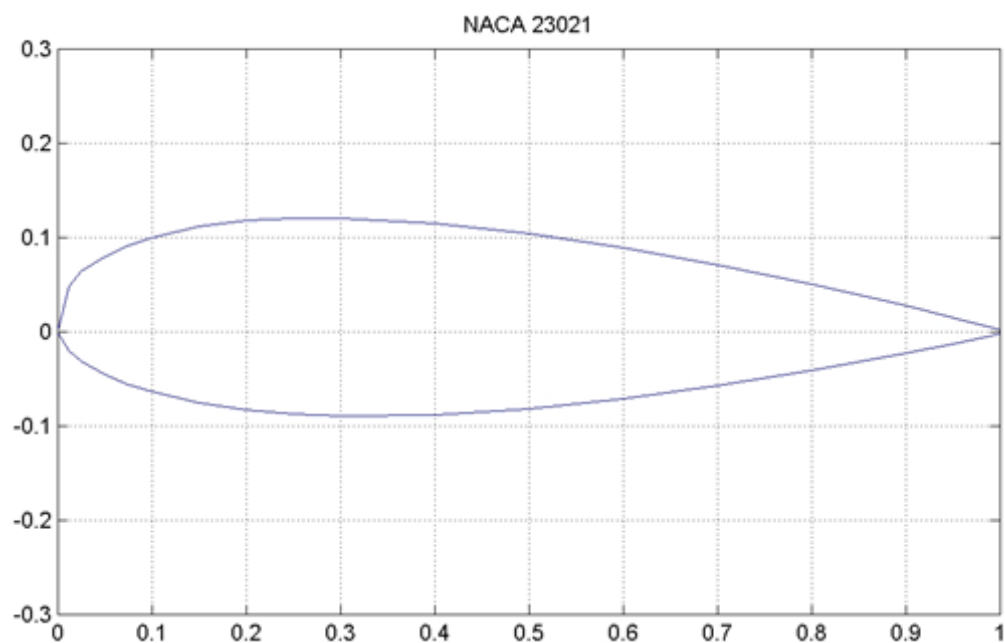
Kolejnym typem skrzydeł ustandaryzowanym przez organizację NACA jest pięciocyfrowa NACA ABCDE, gdzie:

- A – przybliżona wartość  $\frac{20}{3}(C_l)_{proj}$ ,  $(C_l)_{proj}$  to przewidywany współczynnik siły nośnej
- BC – współrzędna strzałki ugięcia, dana wzorem  $200x_m/c$
- DE - relatywna grubość profilu podana w procentach

Przykładowe profile pięciocyfrowe przedstawiliśmy na rysunkach Rys. 2.29 oraz Rys. 2.30



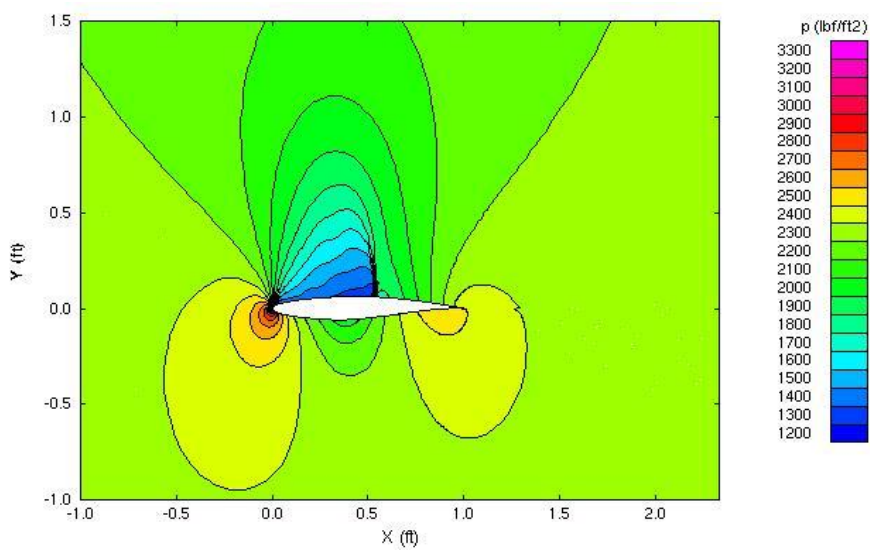
Rys. 2.29 Profil NACA 23015



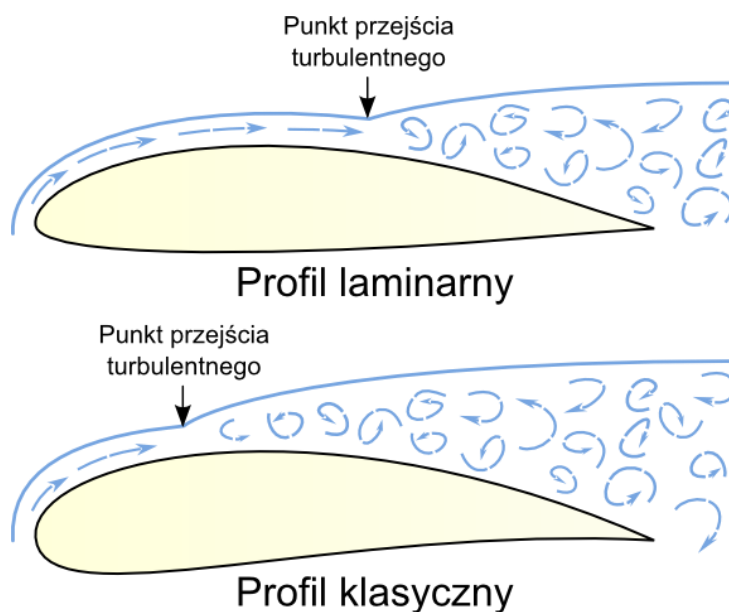
Rys. 2.30 Profil NACA 23021

Przejdziemy teraz do innego typu profili – mianowicie profili o liniowym przepływie. Z powodu występowania warstwy granicznej oraz niezerowej lepkości powietrza pamiętać należy, że dla odpowiednio dużych prędkości (dla prędkości przelotowych samolotów, szybowców) dochodzi do zamiany przepływu liniowego w turbulentny. Ma to oczywiście niekorzystny wpływ na wydajność profilu, gdyż zwiększa się siła oporu tarcia.

Głównym czynnikiem sprzyjającym zmianie przepływu liniowego w turbulentny jest dodatni gradient ciśnienia wzdłuż linii przepływu (zgodnie z równaniem Bernoulliego oznacza to spadek prędkości cząstek). Poruszające się cząstki powietrza napotykać przed sobą wolniejsze. W efekcie może dojść nawet do przepływu „pod prąd” – powstają wiry – i tak dochodzi do zamiany typu warstwy granicznej. Patrząc na dystrybucję ciśnienia  $p$  na powierzchni liniowego profilu lotniczego (**Błąd! Nie można odnaleźć źródła odwołania.**) (ilustracja jest wynikiem symulacji numerycznej) zauważyć można, że minimum (tzw. *suction peak*) znajduje się w połowie cięciwy skrzydła i w okolicach tego miejsca dochodzić będzie do zmiany typu przepływu, w przeciwieństwie do tradycyjnego profilu, gdzie zmiana taka następuje znacznie bliżej krawędzi natarcia skrzydła. W schematyczny sposób efekt ten przedstawia kolejna ilustracja (Rys. 2.32).

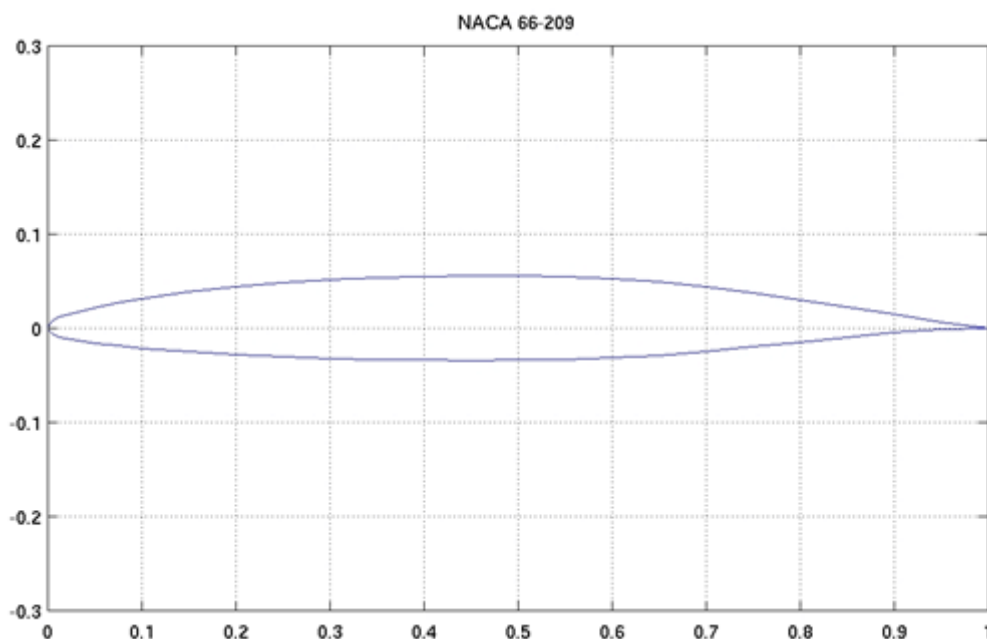


Rys. 2.31 Rozkład ciśnienia na powierzchni profilu liniowego



Rys. 2.32 Porównanie przepływu wokół zwykłego i liniowego profilu

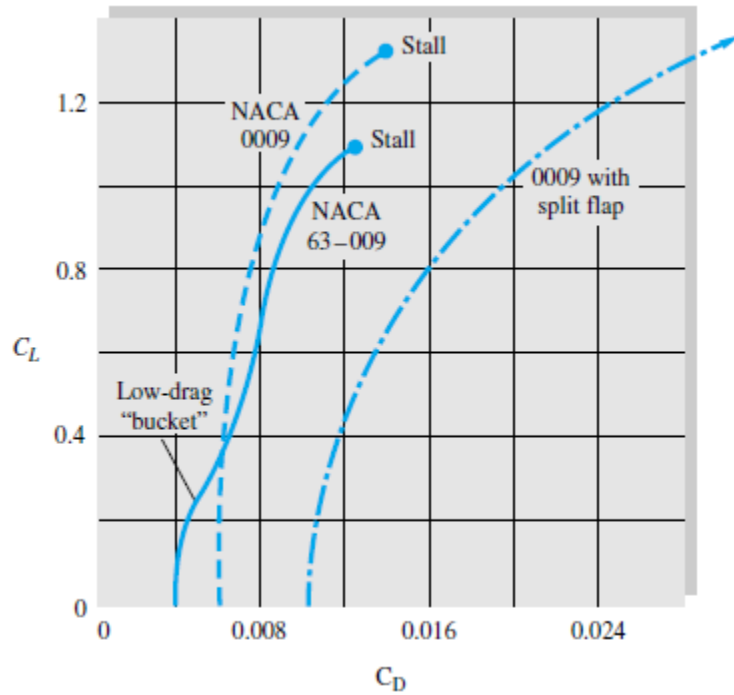
Następny rysunek (Rys. 2.33) przedstawia ostatni – sześciocyfrowy typ profilu NACA. Właśnie ten typ zarezerwowany jest dla profili liniowych.



Rys. 2.33 Profil liniowy NACA 66-209 ( $\alpha=1$ )

Zastosowanie profilu o liniowym przepływie skutkuje zmniejszeniem oporu przepływu w pewnym przedziale prędkości i kątów natarcia. (tzw. *Laminar bucket*). Jest to głównym ograniczeniem przy stosowaniu tego rodzaju profili, zwłaszcza że poza tym obszarem ich właściwości bywają gorsze niż tradycyjnych profili. Kolejnym ograniczeniem jest konieczność zachowania gładkiej powierzchni. Co ciekawe pierwszy raz profil liniowy zastosowano w amerykańskim samolocie Mustang z okresu II wojny światowej, jednak niedostatki w technologii nie pozwoliły uzyskać wystarczająco gładkiej powierzchni. W efekcie profil ten wcale nie działał tak, jak życzyliby sobie konstruktorzy – z powodu nierówności przepływ turbulentny zaczynał się o wiele bliżej krawędzi natarcia. Profile liniowe są powszechnie stosowane przy konstrukcji elektrowni wiatrowych.

Poniżej (Rys. 2.34) porównany jest współczynnik  $C_D$  i  $C_L$  profilu tradycyjnego i liniowego.

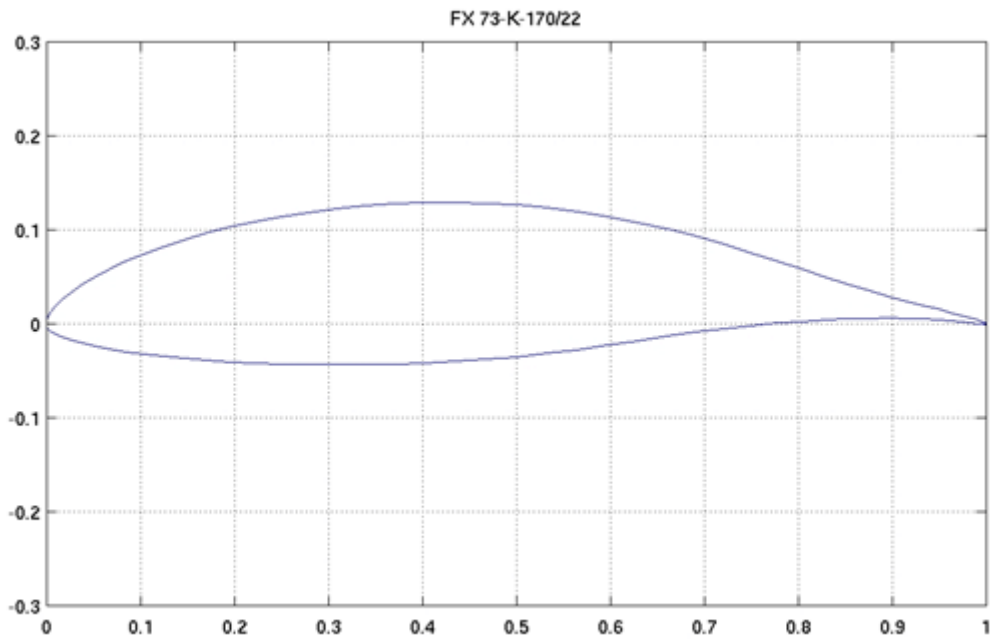


Rys. 2.34 Zależność  $C_L$  ( $C_D$ ) dla profilu liniowego (NACA 63-009), tradycyjnego (NACA 0009) oraz tradycyjnego z otwartymi klapami (0009 with split flap).

Na samym końcu tego rozdziału pokażemy profil profilu Wortmann FX-73-K-170/22 użytego przy budowie szybowca DG-500. Tutaj również przedstawimy wykresy współczynnika siły nośnej, współczynnika oporu oraz wykres zależności  $C_L(C_D)$ .

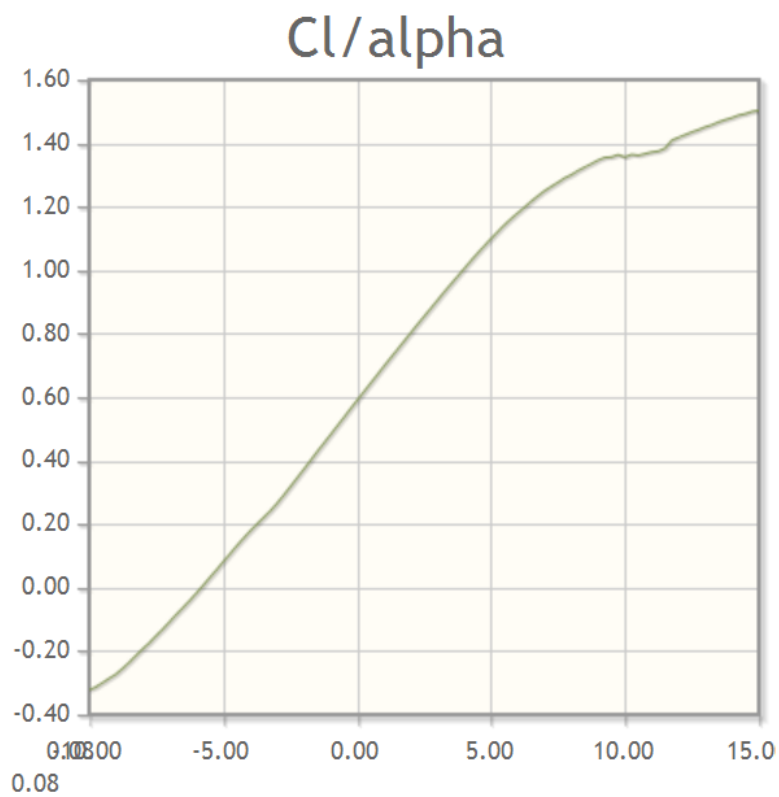
Dane konstrukcyjne dotyczące profilu FX-73-K-170/22:

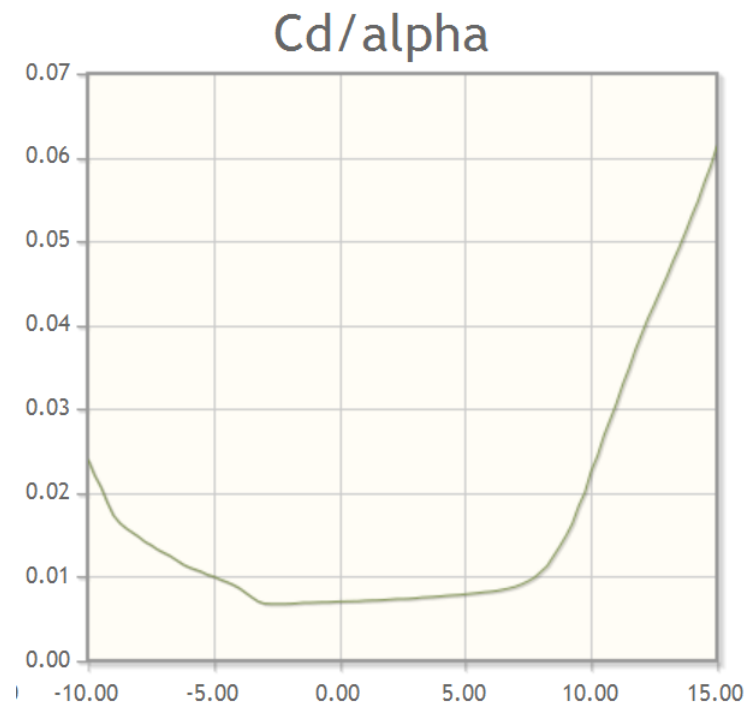
- Maksymalna grubość 17.1% c, umiejscowione w 40.2% cięciwy
- Strzałka ugięcia 4.6% w 56.5% cięciwy



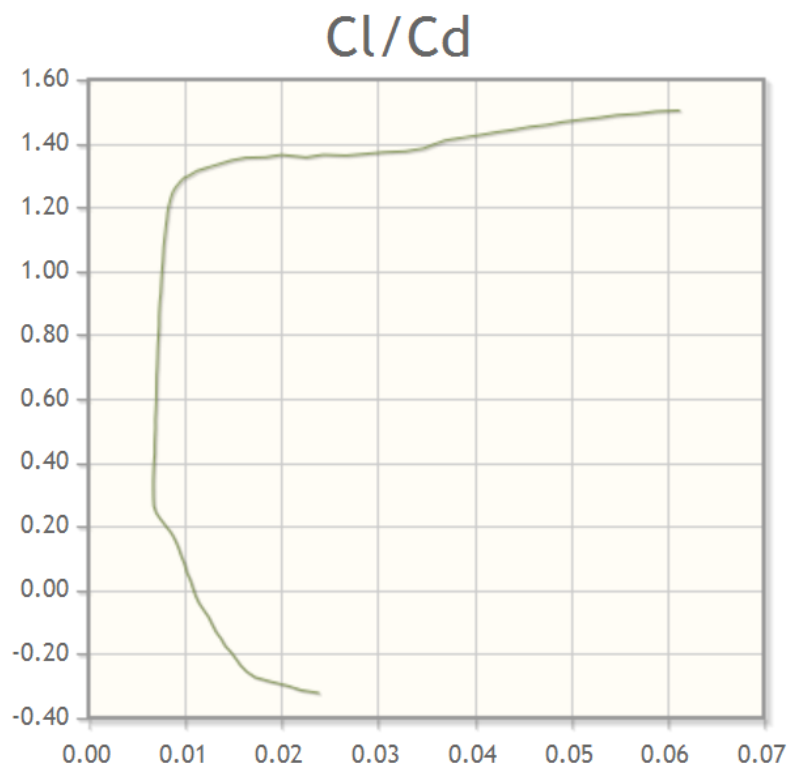
Rys. 2.35 Profil FX 73-K-170/22

Poniższe wykresy współczynników pochodzą z programu do projektowania i analizy profili lotniczych dla przepływów poddźwiękowych *XFOIL*.

Rys. 2.36 Zależność współczynnika siły nośnej  $C_l$  od kąta natarcia  $\alpha$



Rys. 2.37 Zależność współczynnika siły nośnej  $C_d$  od kąta natarcia  $\alpha$

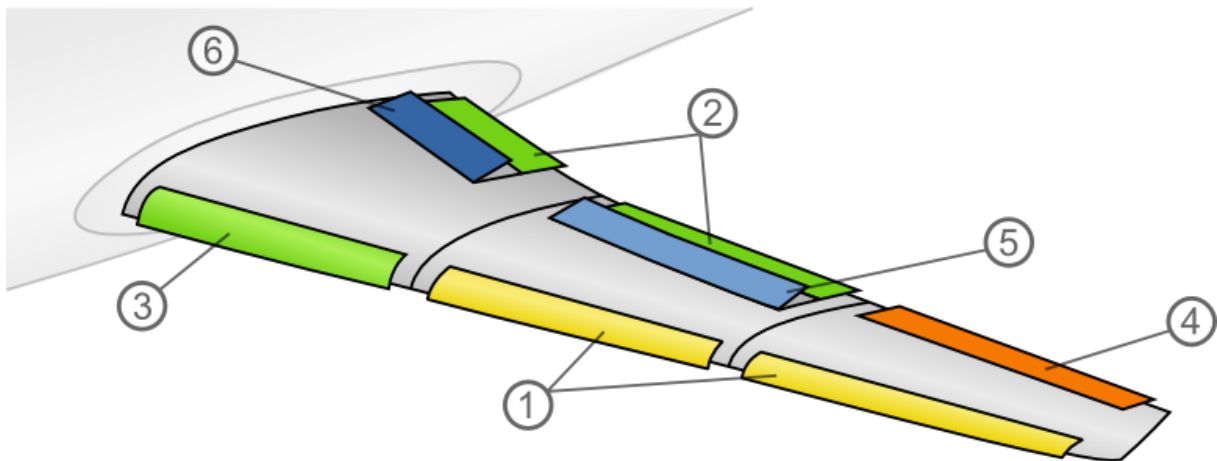


Rys. 2.38 Zależność wartości współczynnika siły nośnej  $C_l$  wartości współczynnika oporu  $C_d$

Na podstawie wykresów tych tworzyć będziemy table danych wejściowych do naszej symulacji.

## 2.2.3 Elementy mechanizacji skrzydła

Ten rozdział poświęcony jest elementom służącym do zmiany niektórych charakterystyk skrzydeł, pożądanym w określonych warunkach lotu. Przede wszystkim chodzi tutaj o dostosowanie skrzydła do pracy w odmiennych warunkach podczas startu i lądowania, kiedy to potrzebna jest bardzo duża siła nośna przy małej prędkości oraz podczas przelotu z prędkością rejsową, kiedy to liczy się zminimalizowanie oporów. Kolejnym istotnym zadaniem niektórych z elementów będzie kontrola warstwy granicznej i zapobieganie wystąpienia przeciągnięcia.



1 – sloty, 2 – klapy, 3 – kłapa Krugera, 4 - lotka, 5 – spojler, 6 – hamulec aerodynamiczny

Rys. 2.39 Elementy mechanizacji skrzydła:

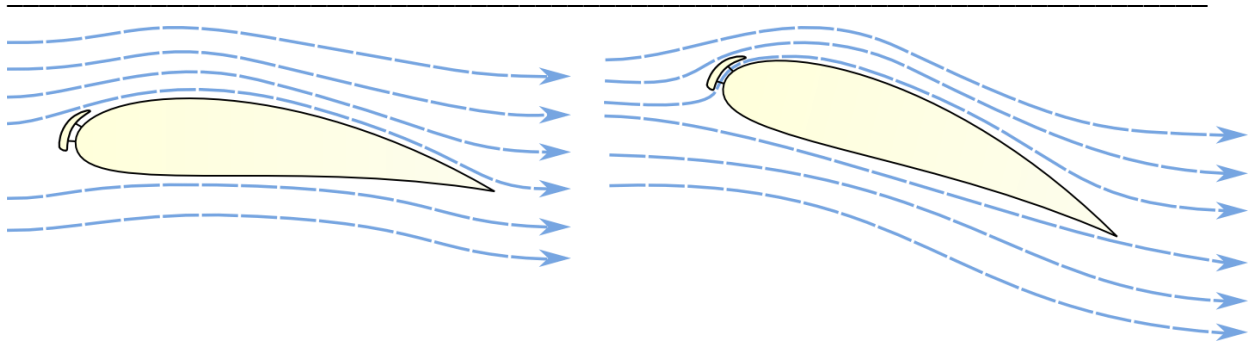


### 2.2.3.1 Sloty

Zwane też skrzelami. Mają za zadanie uniknięcie oderwania strug podczas lotów z dużymi kątami natarcia i z małymi prędkościami. Umieszczane są wzdłuż krawędzi natarcia szybowca lub samolotu. Zastosowanie takiego elementu powoduje zwiększenie cięciwy skrzydła, może również zwiększać jego ugięcie. Dodatkowo, jeśli jest to slot szczelinowy, powietrze z dolnej powierzchni skrzydła zostaje nadmuchiwane na jego górną część, zmniejszając prędkość oderwania strug. Wyróżniamy różne rodzaje slotów:

- automatyczne – wysuwają się z położenia krawędzi natarcia tylko przy spadkach siły nośnej
- nieruchome – na stałe wysunięte
- zasilane – ich położenie może być kontrolowane przez pilota

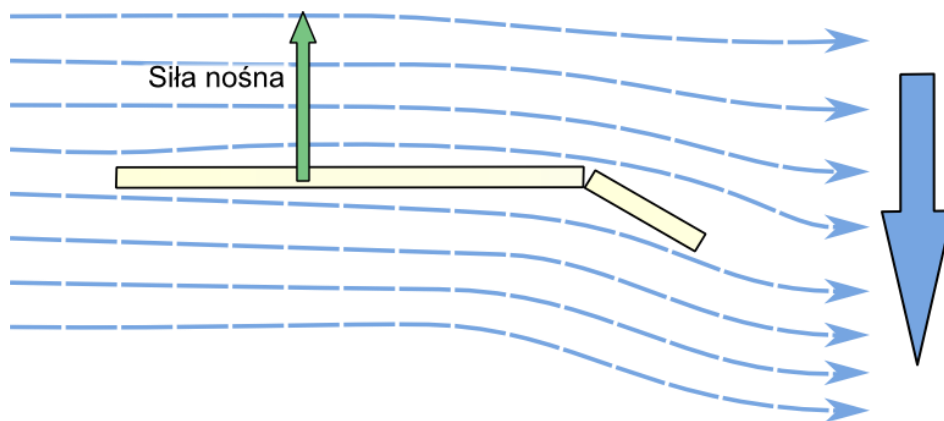




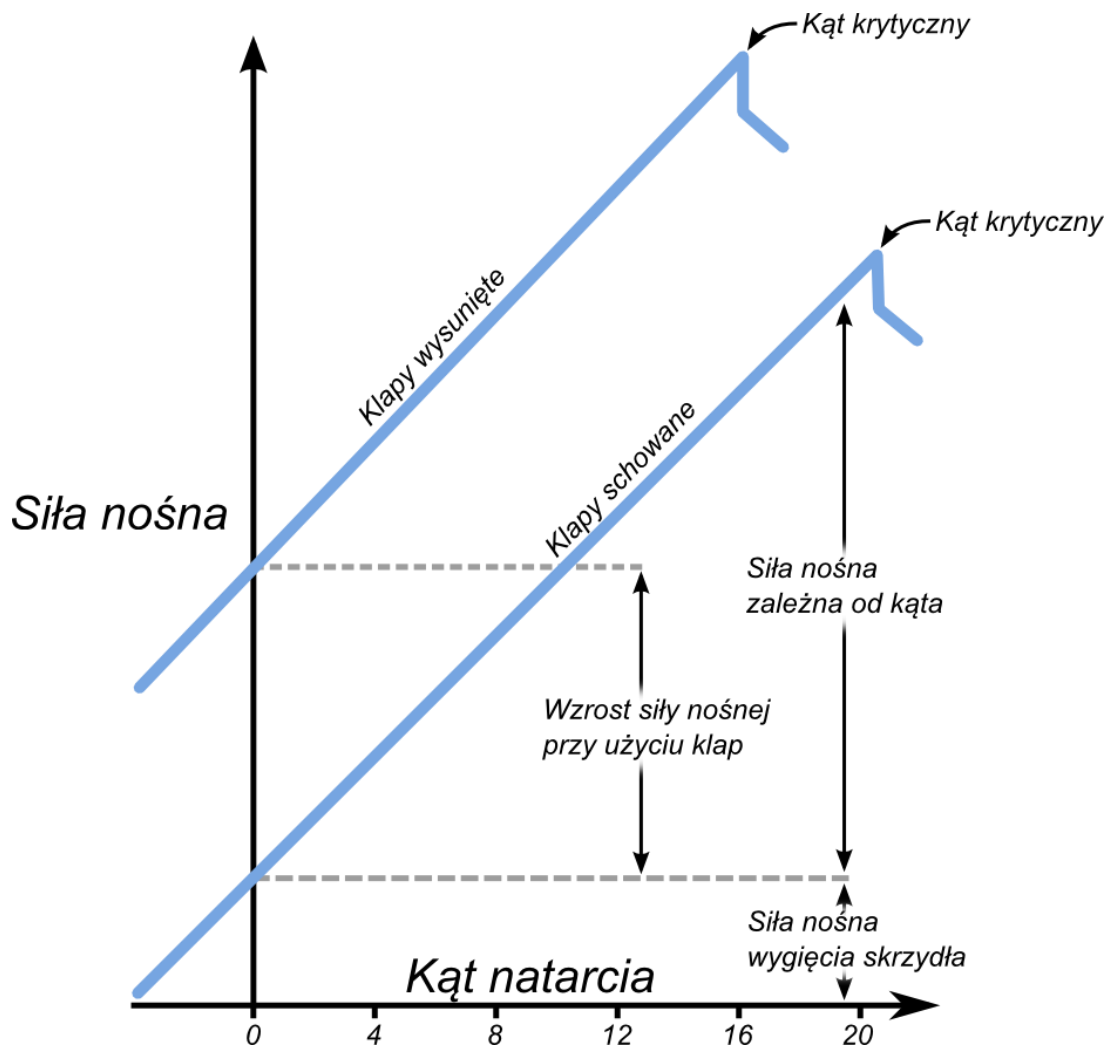
Rys. 2.40 Zasada działania slotu szczelinowego. W trakcie wychylenia skrzydła, pod dużym kątem natarcia, powietrze z dolnej części przedostaje się na górną powierzchnię skrzydła.

### 2.2.3.2 Klapy

Głównym zastosowaniem klap jest utrzymanie stałego poziomu lotu przy niskiej prędkości szybowca. Zwalniający szybowiec wytraca znaczną wartość siły nośnej, aby zaistniałe braki nadrobić pilot może zwiększyć kąt natarcia pod jakim skierowane są skrzydła. Jednak przy ekstremalnie niskich prędkościach wymagany kąt natarcia może przekraczać kąt krytyczny. W takiej sytuacji pomocne okazują się klapy, czyli ruchome końcówki skrzydła znajdujące się w jego tylnej części. Powodują one ponowne odbicie strug powietrza przepływających pod skrzydłem (Rys. 2.41). Klapy przeważnie montowane są na wewnętrznej części skrzydła i wychylają się nie więcej niż 30 stopni od prostej wyznaczonej przez jego cięciwę. Znajdują się na obu skrzydłach i zawsze przyjmują tę samą pozycję, co powoduje symetryczne zwiększanie siły nośnej. Wadą stosowania klap jest spore obniżenie wartości kąta, przy którym następuje oderwanie strug, przez wprowadzenie drugiego dodatniego gradientu ciśnienia (Rys. 2.42).



Rys. 2.41 Wpływ klap na powstawanie siły nośnej na płaskiej powierzchni dla zerowego kąta natarcia. Gdyby nie klapy, siła nośna byłaby równa zero (Rys. 2.2c), jednak wprowadzenie ugięcia wpływa na kształt strug powodując powstanie siły nośnej

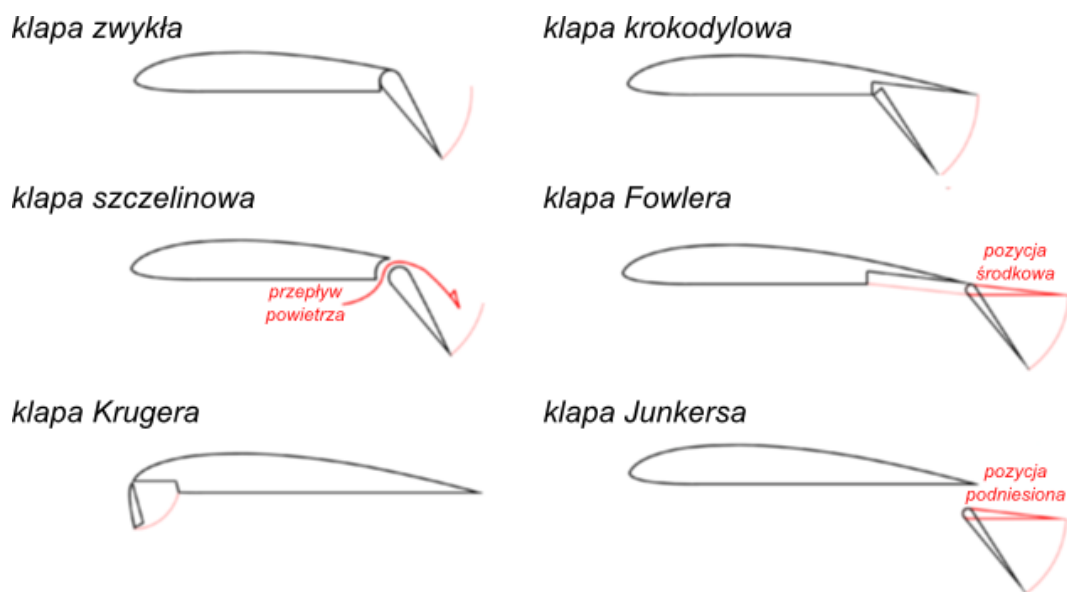


Rys. 2.42 Przykładowy efekt użycia kłap. Widzimy, że siła nośna jest o wiele większa, lecz niestety jest to opłacone niższym kątem krytycznym

Istnieje wiele rodzajów kłap (Rys. 2.43). Te najczęściej stosowane to:

- zwykłe – wychylane w dół, ze względu na lepsze osiągi kłap innego typu stosowane są głównie w sytuacjach gdy stawia się na prostotę wykonania. Przy większych wychyleniach wytwarzają duże ilości oporu
- krokodylowe – tylko dolna część tylnej część skrzydła wychlana jest w dół, górna część pozostaje nieruchoma, konstrukcja mało wydajna, która szczególnie przy większych wychyleniach cechuje się generowaniem większej ilości oporu niż siły nośnej
- szczelinowe – charakteryzuje się występowaniem szczeliny między końcem nieruchomej części skrzydła oraz początkiem kłapy. Dzięki takiej konstrukcji umożliwia napływanie powietrza z dolnej części skrzydła, co podobnie jak w przypadku slotów zmniejsza ilość odrywanych strug

- Fowlera – jest to wariant klapy krokodylowej, w której ruchoma część montowana jest na szynach. Przy obniżaniu kłapa przesuwa się do tyłu zwiększając nie tylko kąt nachylenia ale także powierzchnię skrzydła
- Junkersa – odmiana klapy szczelinowej, od której odróżnia ją fakt, że umieszczona jest za krawędzią spływu. Cechuje ją wytwarzanie większego oporu w pozycji spoczynkowej, lecz zarazem o wiele skuteczniej wytwarza siłę nośną gdy jest używana
- Krugera – kłapa przednia. Wysuwana jest z dolnej części skrzydła znajdującej się zaraz za krawędzią natarcia. Powoduje zwiększenie wygięcia oraz szerokości skrzydła, a co za tym idzie zarówno siły nośnej jak i tarcia



Rys. 2.43 Niektóre rodzaje kłap

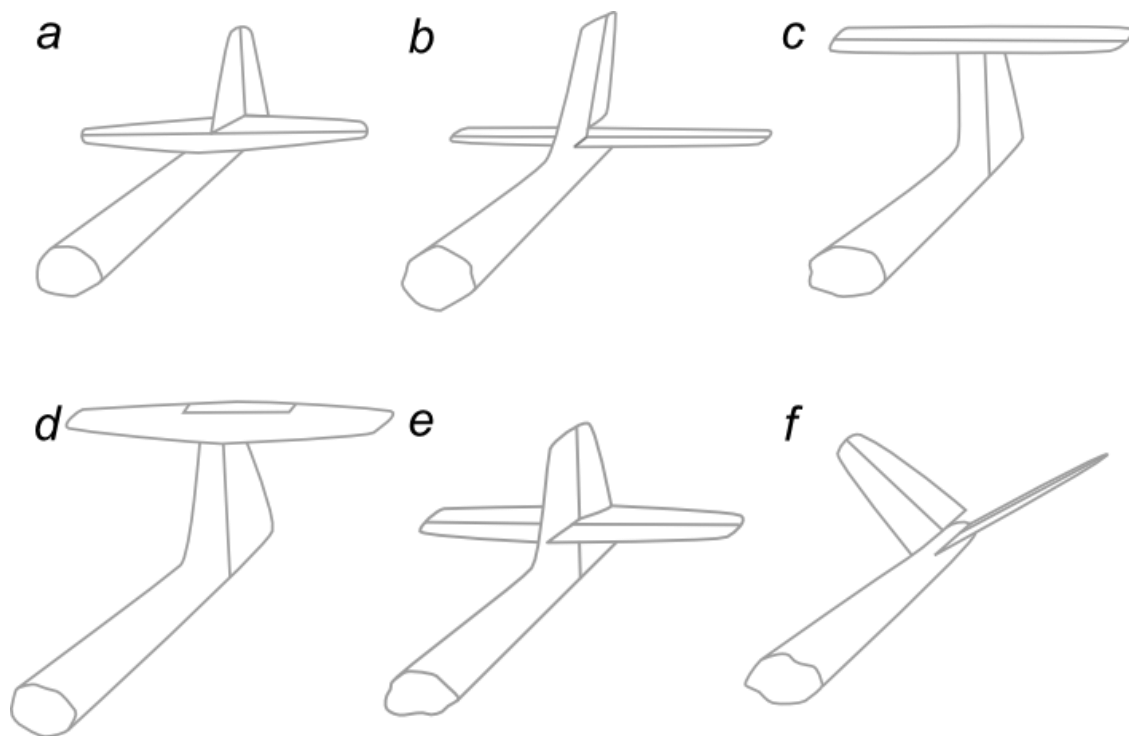
### 2.2.3.3 Lotki

Lotki w konstrukcji przypominają klapy jednak mają zupełnie odmienne zastosowanie. Są one montowane przeważnie na zewnętrznej części skrzydeł i od kłap odróżnia je to, że poruszają się w przeciwną stronę. Obniżaniu lotki na jednym skrzydle, co powoduje wzrost siły nośnej towarzyszy jednocześnie podnoszenie lotki na drugim skrzydle, co powoduje spadek siły nośnej. Powstały w ten sposób nierówny rozkład sił na skrzydłach stwarza możliwość obracania się i wykonywania manewrów.

### 2.2.3.4 Usterzenie

Do elementów mechanizacji skrzydła zaliczamy również stery będące częściami usterzenia szybowca. Przez usterzenie rozumie się jego całość, a więc części nieruchome, zwane statecznikami, oraz części ruchome, zwane właśnie sterami. Usterzenie przeważnie znajduje się w tylnej części kadłuba szybowca. Wyróżniamy dwa rodzaje sterów:

- ster wysokości - ruchomy fragment tylnej części statecznika poziomego. Wychylenie steru wysokości powoduje obrót wokół osi poprzecznej, a co za tym idzie zmianę kierunku lotu oraz kąta natarcia
- ster kierunku - Ruchomy fragment tylnej części statecznika pionowego. Wpływa na obrót samolotu wokół osi pionowej i wraz z odpowiednio zsynchronizowanym ruchem lotek służy do wykonywania zakrętów



Rys. 2.44 Rodzaje usterzenia szybowca

Usterzenia szybowców budowane są w wielu układach charakteryzujących się odmiennymi właściwościami aerodynamicznymi (Rys. 2.44). Często stosowanymi rozwiązaniami są układy ze sterem wysokości umieszczonym na wysokości osi wzdłużnej kadłuba przed sterem kierunku (Rys. 2.44a) lub za sterem kierunku (Rys. 2.44b). To drugie rozwiązanie pozwala uprościć konstrukcje dzięki możliwości zastosowania jednoczęściowego steru wysokości, jednocześnie utrudniając użytkowanie gdyż ster wysokości znajduje się bardzo nisko i wzrasta możliwość jego uszkodzenia. W celu pozbycia się tych niedogodności stworzone zostało tzw. usterzenie typu „T” (Rys. 2.44c,d). Dzięki umieszczeniu steru wysokości nad sterem kierunku jest on mniej narażony na uszkodzenia przy lądowaniu. Minusem takiego rozwiązania jest jednak duża odległość steru wysokości od osi wzdłużnej kadłuba, co powoduje powstawanie na tym sterze większych momentów. Konstrukcja taka musi być więc bardziej wytrzymała. Układem będącym kompromisem między przedstawionymi wcześniej jest układ z usterzeniem wysokości znajdującym się mniej więcej w połowie steru kierunku (Rys. 2.44e). Wymaga to jednak zastosowania

dwupłatowego steru kierunku. Układem zasługującym na szczególną uwagę jest usterzenie motylkowe (Rys. 2.44f). Zostało ono wynalezione przez polskiego inżyniera Jerzego Rudnickiego w 1930 roku. Charakteryzuje się zastąpieniem statecznika pionowego oraz poziomego ( a co za tym idzie sterów kierunku oraz wysokości) dwoma statecznikami ustawionymi do siebie pod odpowiednim kątem, które formują układ w kształcie litery „V”. Powierzchnie sterowe zamontowane na tych statecznikach w zależności od wychylenia mogą więc spełniać zarówno rolę steru kierunku jak i wysokości. Zaletami takiego usterzenia jest bardzo ważna w szybownictwie lekkość konstrukcji oraz zmniejszenie ilości powstających zawirowań. Minusem natomiast jest bardziej złożony układ sterowniczy oraz wytwarzanie większych naprężeń przy sterowaniu.

### 2.2.3.5 *Hamulce aerodynamiczne*

Element mechanizacji skrzydła służący do wytwarzania dużych wartości oporu. Wykorzystywany głównie przy lądowaniu, gdy potrzebny jest nagły spadek prędkości statku powietrznego, lub też chwilowe pogorszenie doskonałości aerodynamicznej. W nowoczesnych, wysokowydajnych szybowcach ich zastosowanie jest bardzo istotne ze względu na uzyskiwane przez nie bardzo dużego stosunku siły nośnej do oporu (wysoka doskonałość aerodynamiczna). Jest to jednak niekorzystne w przypadku lądowania, ponieważ bez użycia hamulców możliwe byłoby jedynie lądowanie z niską prędkością i bardzo małym kątem ścieżki opadania, lub też zwiększenie kąta ścieżki opadania po skierowaniu dzioba w stronę ziemi, co jednak spowodowałoby znaczny wzrost prędkości. Problem ten rozwiązuje zastosowanie hamulców aerodynamicznych zwiększających opór.



Rys. 2.45 Schematyczny rysunek szybowca z wysuniętymi hamulcami aerodynamicznymi

### 2.2.3.6 *Spoilery*

Posiadają podobne zastosowanie jak hamulce aerodynamiczne. Różnią się jednak tym, że po wychyleniu poza wzrostem oporu gwarantują także znaczny spadek siły nośnej, poprzez wytwarzanie siły nośnej skierowanej w przeciwnym kierunku. Spoilery są bardzo często wykorzystywane w motoryzacji gdzie przy dużych prędkościach jazdy zapewniają lepszy docisk kół do podłoża.



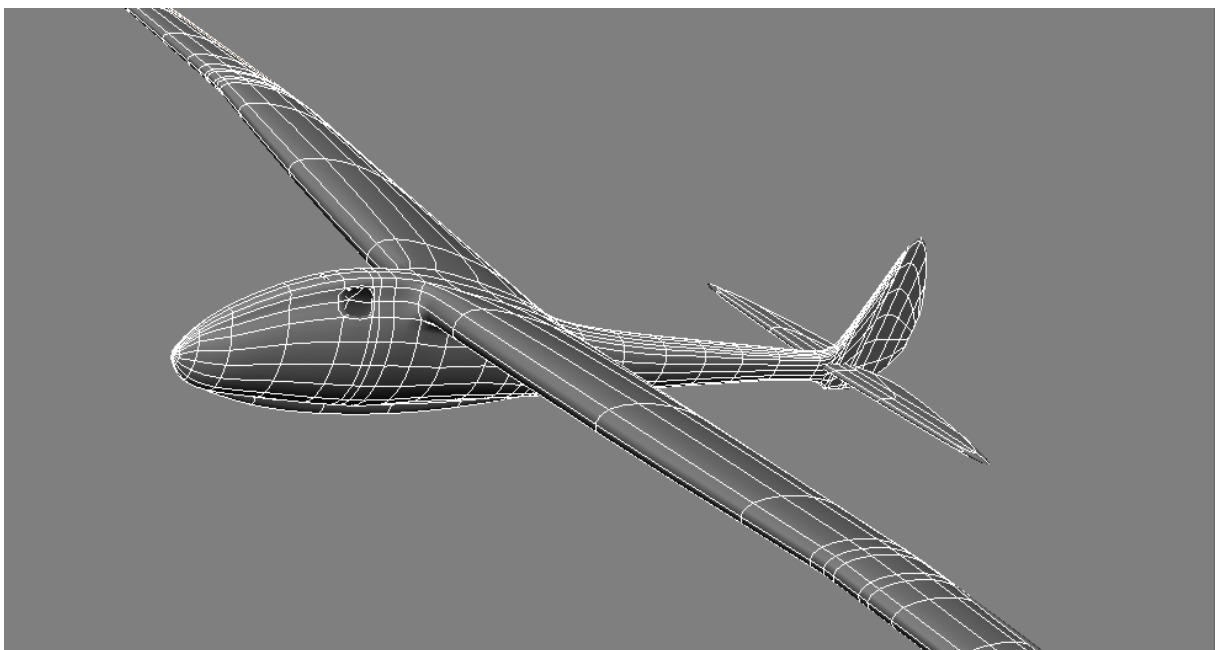
a) spojler w samolocie pasażerskim

b) spojler w samochodzie wyścigowym

Rys. 2.46 Przykłady spojlerów

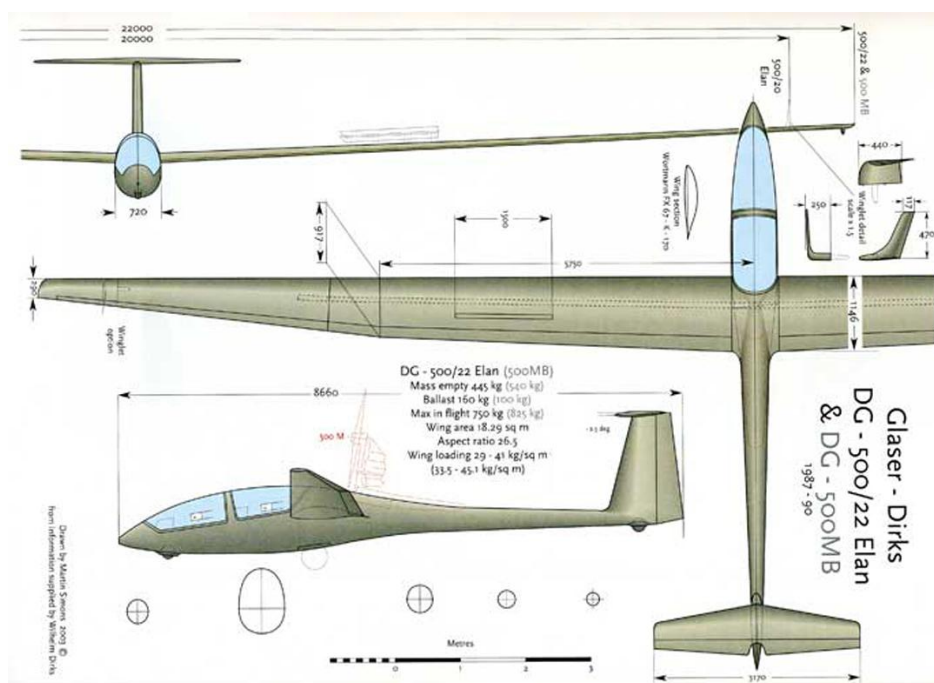
## 3 Konstrukcja modelu

W rozdziale tym zajmiemy się omówieniem procesu tworzenia modeli szybowców oraz świata, w którym będą się one poruszać. Modele te stworzone zostały w środowisku *Autodesk Softimage*, które dzięki ogromnym możliwościom i rozbudowanej funkcjonalności jest doskonałym narzędziem do tworzenia nowoczesnej grafiki komputerowej. W pierwszej kolejności zajmiemy się opisem powstawania obiektów geometrycznych szybowców. Następnie omówimy zastosowane materiały odwzorowujące wygląd rzeczywistych materiałów stosowanych przy konstrukcji szybowców. W tym momencie wprowadzimy również nieodzowny element stosowania materiałów jakim jest oświetlenie sceny, w którym zawrzemy również definicję słońca oraz nieboskłonu naszej sceny. Model konstruowanego otoczenia składać się będzie także z terenu, nad którym przemieszczać będą się nasze szybowce. W dalszej części rozdziału przedstawimy więc sposób tworzenia geometrii terenu. W tym wypadku zamiast konwencjonalnych narzędzi modelarskich programu *Softimage* posłużymy się środowiskiem *ICE (Interactive Creative Environment)* służącym do tworzenia zautomatyzowanych systemów proceduralnych mogących służyć szerokiej gamie zastosowań w tym przetwarzaniu geometrii elementów sceny. Sposób symulacji utworzonych modeli omówiony zostanie w rozdziale następnym.



## 3.1 Model szybowca

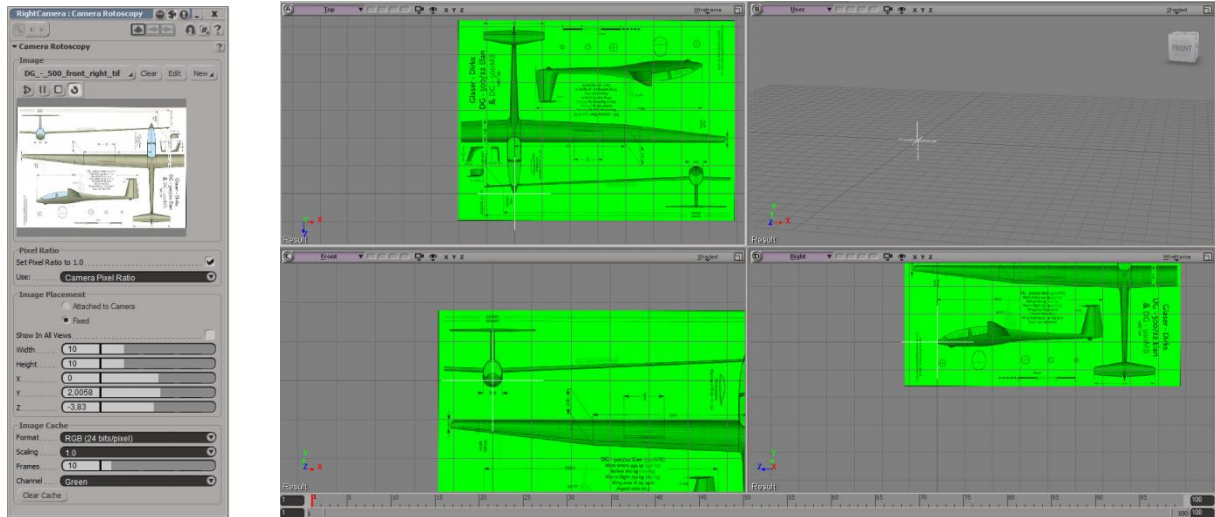
### 3.1.1 Geometria modelu



Rys. 3.1 Plany szybowca *Glaser-Dirks DG-500*

Zajęliśmy się konstrukcją modelu szybowca *Glaser-Dirks DG-500*. Podczas konstrukcji posłużyliśmy się poglądowymi planami szybowca z książki *Sailplanes 1965-2000* (Rys. 3.1). Jak widać plany te prezentują szybowiec z trzech perspektyw - od góry, z przodu oraz z boku. Dzięki takiej prezentacji szybowca jesteśmy w stanie wyeksportować plan na odpowiednie płaszczyzny sceny 3D. W tym celu korzystamy z tzw. rotoskopu, które to narzędzie służy do umieszczania obrazów w tle wybranych widoków sceny. Okno dialogowe rotoskopu (Rys. 3.2a) pozwala nam ustawić parametry dodawanego obrazu takie jak rozmiar (wysokość i szerokość), których wartości pozostawiamy domyślne, a także przesunięcie obrazu na płaszczyźnie. Parametry te (X, Y, Z) dostosowujemy w taki sposób, aby w każdym z trzech widoków sceny dziób szybowca widzianego z odpowiedniej perspektywy znajdował się w początku układu współrzędnych (Rys. 3.2b). Dodatkowo w opcjach *Image Cache* ustawiamy, aby wyświetlany był tylko jeden kanał obrazu. Pozwala to w szybki sposób zmienić reprezentację tła na takie, które zdecydowanie mniej zlewa się kolorystycznie z elementami umieszczanymi na scenie. Odpowiednie ustawienie rotoskopu pozwoli nam w bardzo dobrym stopniu odwzorować krzywizny szybowca, co spowoduje utworzenie realistycznie odwzorowanego modelu.

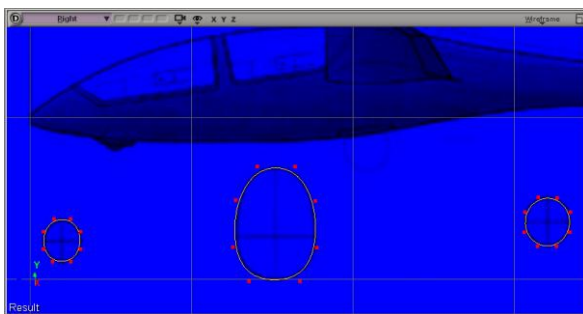




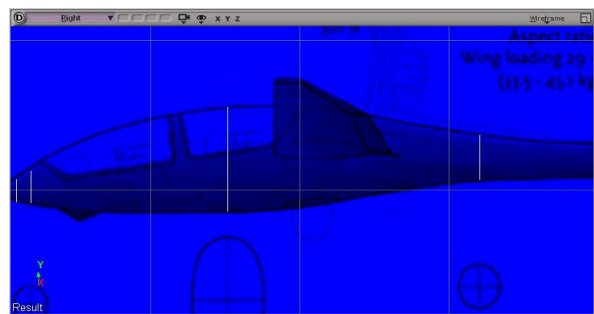
a) przykładowe ustawienie rotoskopu

b) wygląd rotoskopów ustawionych w trzech widokach

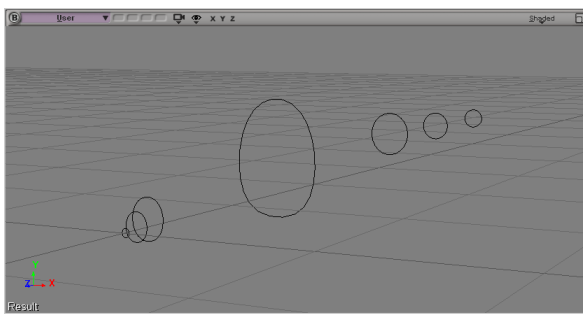
Rys. 3.2 Rotoskop modelu w środowisku Softimage



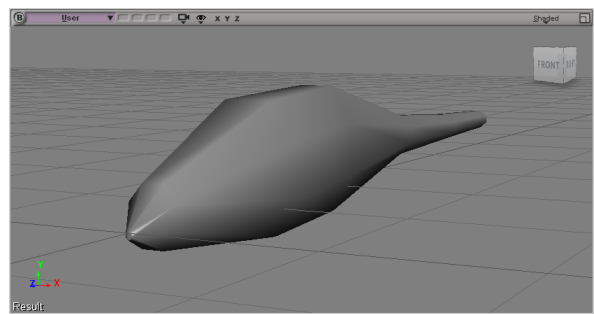
a) tworzenie okręgów



b) odpowiednie ustawienie na scenie 3D



c) widok z perspektywy



d) zastosowanie funkcji loft

Rys. 3.3 Tworzenie okręgów oraz zastosowanie funkcji loft

Po umieszczeniu rotoskopu zajęliśmy się obrysowywaniem kształtu kadłuba. W tym celu wykorzystaliśmy okręgi utworzone przy pomocy krzywych Béziera (patrz Krzywa Béziera). Okręgi te następnie połączyliśmy geometrią wykorzystując funkcję *Polygon Mesh* → *Loft* (patrz Zastosowane narzędzia). Podczas tworzenia modelu przyjęliśmy korzystną dla optymalizacji złożoności obiektów strategię zaczynania od najmniejszej możliwej geometrii. Wystarczająco dobre odwzorowanie elipsoidalnych kształtów kadłuba otrzymaliśmy poprzez obrysowanie ich okręgami składającymi się z sześciu punktów (Rys. 3.3a). Następnie utworzone okręgi ustawione zostały w odpowiednim miejscu na scenie 3D (Rys. 3.3b, Rys. 3.3c). Ostatnim krokiem tworzenia geometrii niskiej rozdzielczości było zastosowanie funkcji *loft* na utworzonych okręgach (Rys. 3.3d). Widzimy, że geometria posiada zdecydowanie zbyt małą złożoność (102 elementy) żeby dobrze odwzorowywała krzywizny kadłuba. W celu poprawienia dokładności modelu utworzona została nowa geometria typu *Subdivision Surface* (patrz *Subdivision Surface*) bazująca na dotychczasowej geometrii (Rys. 3.5). Zauważyć możemy, że poprzez zastosowanie podziału stopnia trzeciego zgodnie z regułą Catmull-Clark zdecydowanie wzrosła ilość trójkątów składających się na obiekt (aż do 6528), co wynika z prostej zależności zgodnej z zastosowaną regułą podziału  $102 \cdot 4^{\text{stopień}} = 102 \cdot 4^3 = 6528$ . Zauważyć możemy też, że kształt geometrii naszego kadłuba wymaga większej złożoności na płaszczyźnie XY (konieczność zachowania gładkiej elipsoidalnej płaszczyzny) niż wzdłuż osi Z. W związku z tym łatwo możemy wyeliminować niepotrzebną geometrię wzdłuż tej osi szczególnie w okolicach ogona gdzie krzywizna wzdłuż osi Z jest praktycznie prostą (Rys. 3.6).

### Krzywa Béziera

Parametryczna krzywa wielomianowa opracowana w latach 60. XX w. niezależnie przez Pierre'a Béziera, francuskiego inżyniera firmy Renault oraz Paula de Casteljaou, pracującego dla koncernu Citroën. Krzywe Béziera ze względu na swoją elastyczność, prostotę oraz możliwość zachowania ciągłości wykorzystywane są w grafice komputerowej przy wyznaczaniu krzywych w przestrzeni, na których podstawie projektowane są kształty figur i powierzchni. Kształt krzywych Béziera opisywany jest wielomianami Bernsteina:

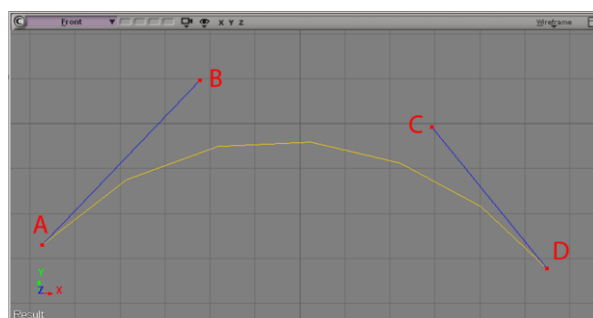
$$p(t) = \sum_{i=0}^n p_i B_i^n(t) \text{ dla } t \in [0,1]$$

$n$  – stopień wielomianu,

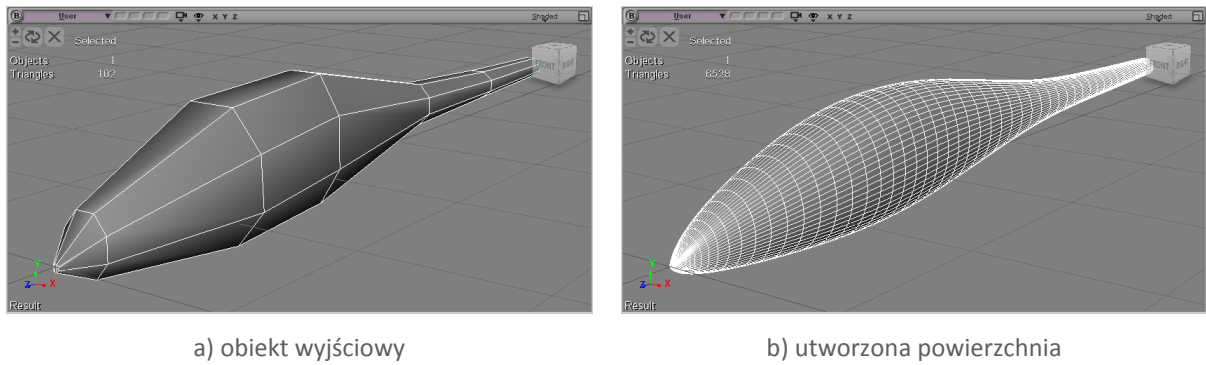
$p_i$  –  $i$ -ty punkt kontrolny krzywej

W programach graficznych wykorzystuje się krzywe Béziera opisane wielomianami trzeciego stopnia, ponieważ charakteryzują się odpowiednią elastycznością i wystarczającą złożonością. Kształt takiej krzywej określany jest czterema punktami kontrolnymi  $A, B, C, D$ :

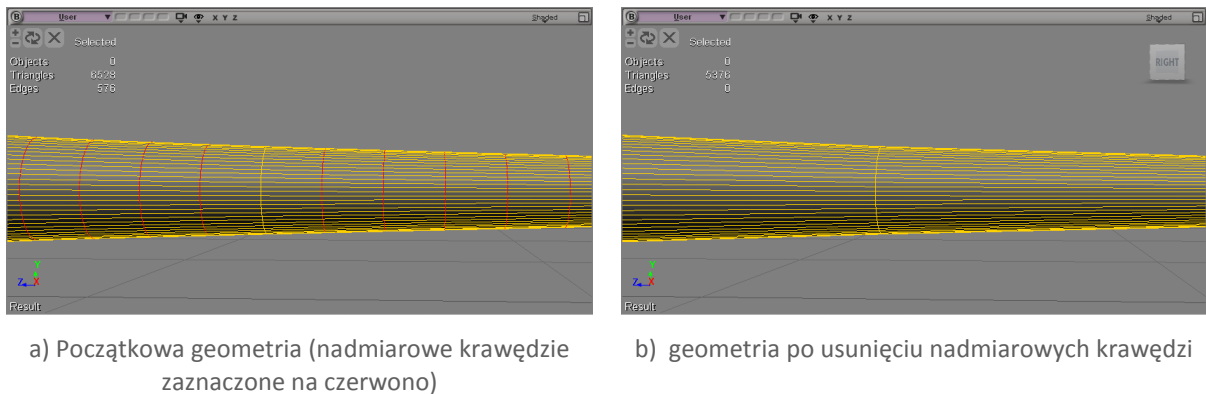
$$P(t) = A(1-t)^3 + Bt(1-t)^2 + 3Ct^2(1-t) + Dt^3 \text{ dla } t \in [0,1]$$



Rys. 3.4 Krzywa Béziera w programie *Softimage*

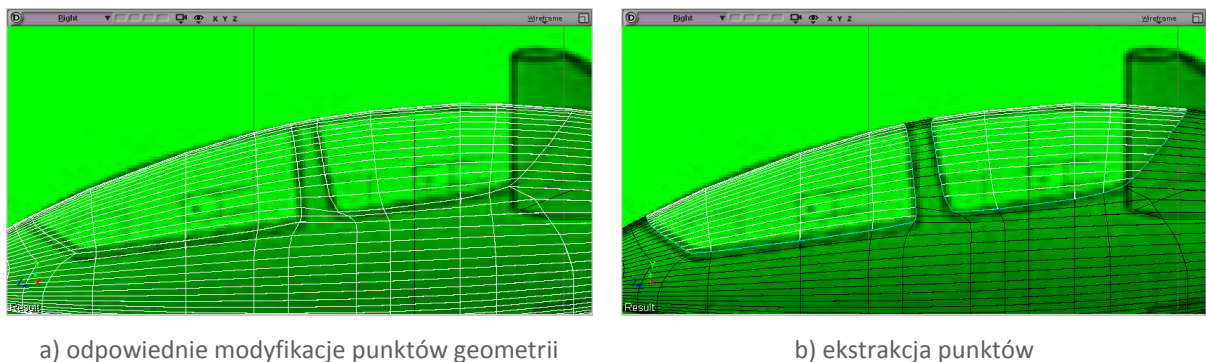


Rys. 3.5 Porównanie obiektu bazowego i utworzonej na jego bazie powierzchni podziału

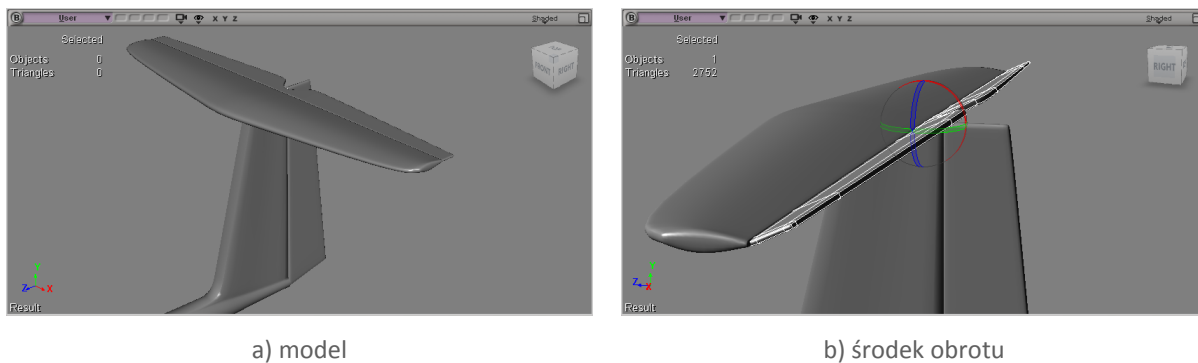


Rys. 3.6 Usuwanie nadmiarowej geometrii z okolic ogona modelu

Następnie zajęliśmy się wycinaniem z kadłuba geometrii znajdującej się w miejscach szyb. W tym celu dokonaliśmy ręcznego przesunięcia odpowiednich wierzchołków, żeby dobrze odwzorować krzywiznę szyby (Rys. 3.7). Następnie wyodrębniliśmy geometrię znajdującą się w miejscu szyb jako nowe obiekty w ten sposób pozbywając się geometrii kadłuba tam gdzie jej być nie powinno jednocześnie otrzymując dwa nowe obiekty które posłużą nam jako szyby.



Rys. 3.7 Modelowanie szyb



a) model

b) środek obrotu

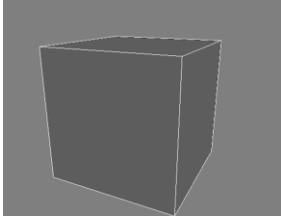
Rys. 3.8 Geometria usterzenia

Przystąpiliśmy do tworzenia usterzenia modelu. W tym celu kontynuując strategię konstruowania elementów z minimalną geometrią wyszliśmy ze zwykłych sześcianów,

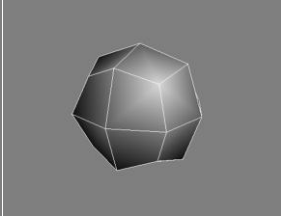
### Subdivison Surface

Jest to rodzaj gładkiej, złożonej geometrii powstałej na bazie siatki wielokątów niskiej rozdzielczości. *Subdivision Surfaces* tworzone są poprzez rekursywne stosowanie wybranej reguły podziału na dostarczonym prototypie. Zastosowanie reguły powoduje podział siatki czemu towarzyszy powstanie nowych węzłów i ścian, których położenie zależy od położenia początkowych elementów oraz wybranej reguły. *Subdivision Surfaces* tworzone są w celu otrzymania modelu o gęstszej siatce co powoduje zwiększenie jego szczegółowości i lepszą jakość, lecz także zwiększa czas renderingu tego elementu. W środowisku *Softimage* dostępne są następujące zasady podziału:

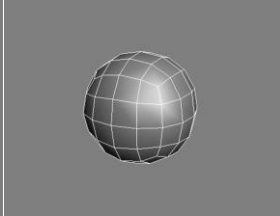
- **Catmull-Clark** – algorytm generuje geometrię silnie wygładzoną składającą się z elementów czworobocznych, która w regionach gdzie oryginalnie były czworoboki posiada ciągłość rzędu  $C^2$ , natomiast w miejscach gdzie występowały elementy nie będące czworobokami lub nie posiadające czterech krawędzi tworzona powierzchnia posiada ciągłość rzędu  $C^1$
- **XSI-Doo-Sabin** – jest odmianą standardowego algorytmu Doo-Sabin tworzącego geometrię wygładzoną. Produkuje więcej geometrii niż algorytm standardowy, ale za to lepiej współpracuje z właściwościami obiektu takimi jak UV mapy, mapy wagowe lub fałdy. Na obszarach gdzie oryginalna topologia składała się z czworościanów, utworzona powierzchnia posiada ciągłość rzędu  $C^1$
- **Linear Subdivision** – reguła ta nie stosuje żadnego wygładzania, co powoduje, że kształt obiektu pozostaje niezmienny
- **Loop Subdivision** – metoda wykorzystująca podziały *Catmull-Clark* oraz liniowy w celu utworzenia obiektu na zasadach podziału *Catmull-Clark* z tą jednak różnicą, że po podziale obszary składające się z elementów trójkątnych nadal będą się z nich składały



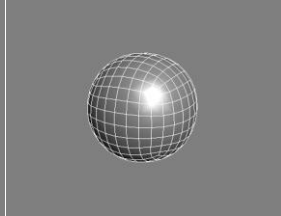
a) geometria początkowa



b) podział – poziom 1



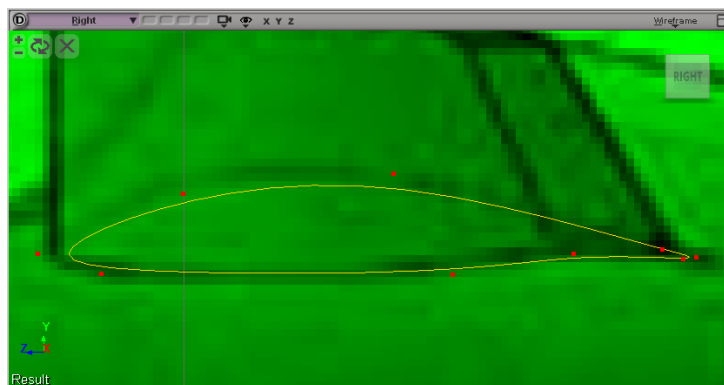
c) podział – poziom 2



d) podział – poziom 3

Rys. 3.9 Zastosowanie podziału Catmull-Clark

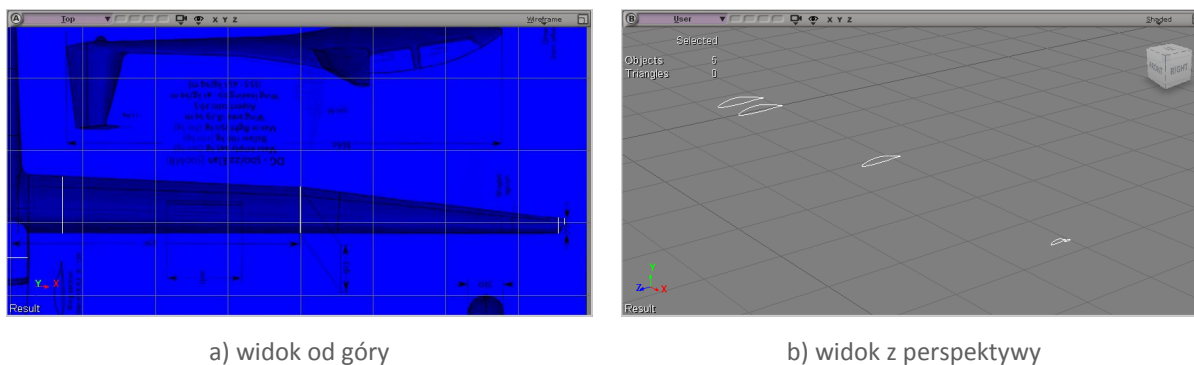
z których następnie w procesie dodawania geometrii z zastosowaniem narzędzia *Add Edge Loop Tool* (patrz Zastosowane narzędzia) otrzymaliśmy elementy z odpowiednim zagęszczeniem w interesujących nas częściach. Końcowy kształt usterzenia (Rys. 3.8a) otrzymaliśmy poprzez odpowiednie przesunięcia wierzchołków posiadanego elementu. Podstawa usterzenia została zmontowana z resztą kadłuba poprzez odpowiednie scalenie z wykorzystaniem narzędzia *Add/Edit Polygon Tool* (patrz Zastosowane narzędzia). Pozostałe elementy usterzenia, którymi są lotka steru kierunku oraz podstawa z lotką steru wysokości pozostawione zostały jako osobne obiekty sceny. Następnie zadaliśmy o odpowiednie ustawienie punktu obrotu elementów ruchomych tak, aby następowały one wzdłuż osi wyznaczonej przez punkty styku z elementami nieruchomymi (Rys. 3.8b).



Rys. 3.10 Krzywa Béziera naśladowująca kształt profilu skrzydła

Zajęliśmy się konstrukcją skrzydła. W pierwszej kolejności utworzyliśmy krzywe profilu skrzydła. Widzimy, że zgodnie z planami (Rys. 3.1) szybowiec posiada profil *Wortmann FX-67-K-170*. W celu dokładnego odwzorowania tego profilu obrysowujemy go krzywą Béziera (Rys. 3.10), która w tym wypadku musi być bardziej złożona niż dla elipsoidalnego kadłuba i składa się z dziewięciu węzłów. Wiemy, że profil skrzydła wzdłuż wydłużenia w konstruowanym modelu nie zmienia się toteż, aby utworzyć krzywe profilu wzdłuż skrzydła kopiujemy utworzoną krzywą po czym skalujemy ją i ustawiamy w przestrzeni 3D zgodnie z posiadanymi planami (Rys. 3.11). Po odpowiednim ustawieniu krzywych stosujemy na nich funkcję *Polygon Mesh*  $\rightarrow$  *Loft*, która analogicznie jak w wypadku kadłuba wypełnia geometrią przestrzeń między poszczególnymi krzywymi. Otrzymany element również posiada bardzo małą rozdzielczość, składającą się z zaledwie 168 elementów (Rys. 3.12a), która niewystarczająco dobrze naśladuje skrzydło. Tworzymy więc na jego podstawie element typu *Subdivision Surface* wykorzystując regułę Catmull-Clark poziomu drugiego. Otrzymujemy w ten sposób geometrię o zdecydowanie większej rozdzielczości – 1688 elementów (Rys. 3.12b). Możemy zaobserwować, że podobnie jak przy tworzeniu kadłuba w skrzydle także mamy sytuację w której większość geometrii wzdłuż jednej osi (tym razem jest to oś y) jest zbędna ponieważ wydłużenie skrzydła, z wykluczeniem zaokrąglonego zakończenia, tworzy linię łamaną (złamaną w jednym miejscu, blisko środka skrzydła), co powoduje że wzdłuż tej osi dla dobrego odwzorowania geometrii wystarczy zostawić jedynie parę krawędzi wierzchołków. Postanowiliśmy się więc pozbyć większości krawędzi (Rys. 3.13). Zastosowanie *Subdivision Surface* powoduje jednak,

że geometria w kierunku pozostałych dwóch osi jest gęsta, co pozwala w dobrym stopniu naśladować krzywiznę profilu skrzydła.

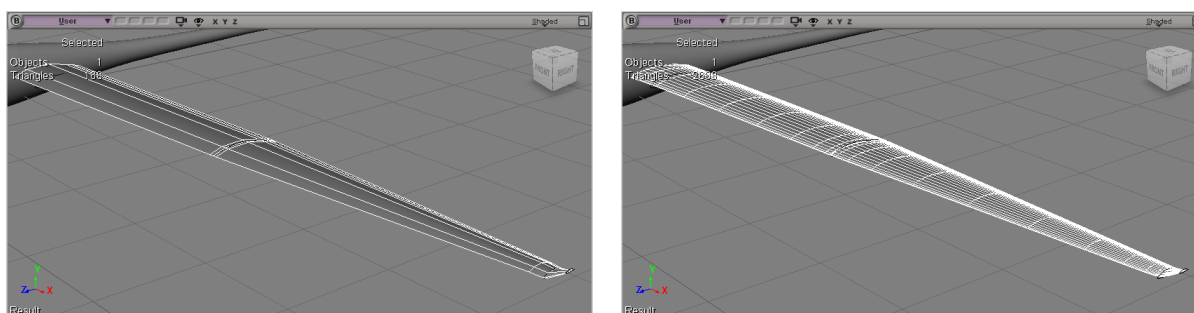


a) widok od góry

b) widok z perspektywy

Rys. 3.11 Utworzone krzywe profilu skrzydła

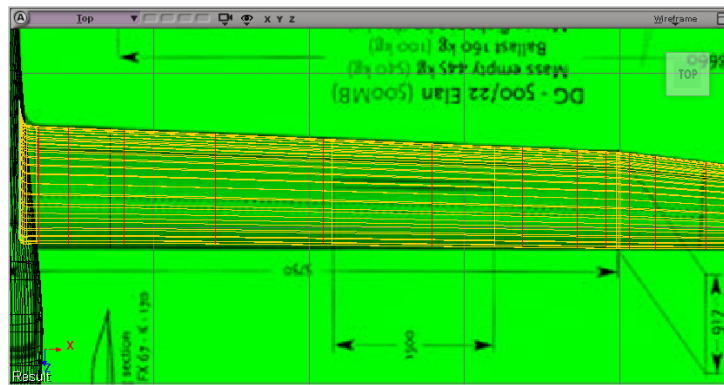
Po otrzymaniu skrzydła z satysfakcjonującą nas ilością geometrii zabraliśmy się za wycinanie z niego części ruchomych tj. hamulca aerodynamicznego oraz lotek. W pierwszej kolejności wycięliśmy hamulec (Rys. 3.14). W tym celu w pierwszej kolejności musieliśmy dodać do geometrii skrzydła krawędzie w miejscach występowania krawędzi hamulca. Wykorzystaliśmy do tego narzędzie *Add Edge Loop Tool*. Następnie wyodrębniliśmy z modelu dwie ściany, po jednej na górze i dole skrzydła, oraz połączyliśmy je ze sobą prostą geometrią używając narzędzia *Bridge* (patrz Zastosowane narzędzia). Otrzymany element jest więc prostym sześcianiem posiadającym jednak odpowiednie rozmiary ścian idealnie pasujące do wyciętej geometrii skrzydła. Analogicznym sposobem zamykamy dziury powstałe w geometrii skrzydła.



a) niska rozdzielczość

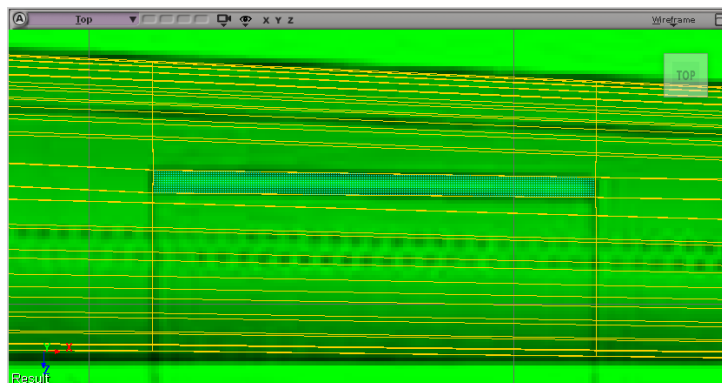
b) wyższa rozdzielczość

Rys. 3.12 Utworzenie skrzydła z posiadanych krzywych oraz zwiększenie rozdzielczości elementu

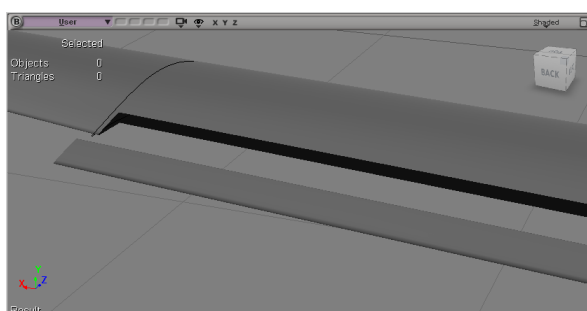


Rys. 3.13 Usuwanie nadmiarowej geometrii ze skrzydła. Krawędzie do usunięcia zostały zaznaczone na czerwono

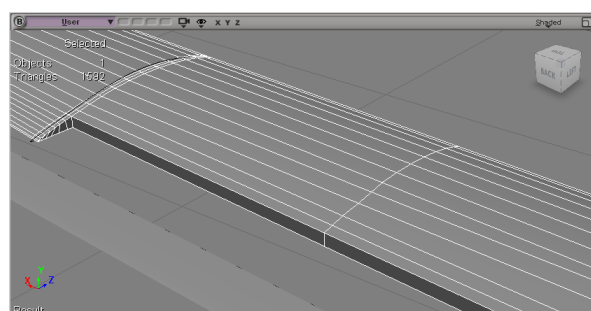
Pozostałymi do wycięcia elementami były lotki. W tym celu, podobnie jak przy tworzeniu hamulca dodaliśmy w odpowiednich miejscach skrzydła geometrię, a następnie wyodrębniliśmy interesujące nas elementy (Rys. 3.15a). Proces tworzenia elementów lotek zakończyliśmy zamykając dziury powstałe zarówno w geometrii skrzydła jak i nowopowstałych obiektach (Rys. 3.15b). Na koniec ustawiliśmy ruchomym elementom środki obrotu w taki sposób, żeby rotowały zgodnie z prostą wyznaczaną przez punkty styczności z nieruchomym elementem skrzydła.



Rys. 3.14 Wyodrębnianie z elementu skrzydła geometrii hamulca



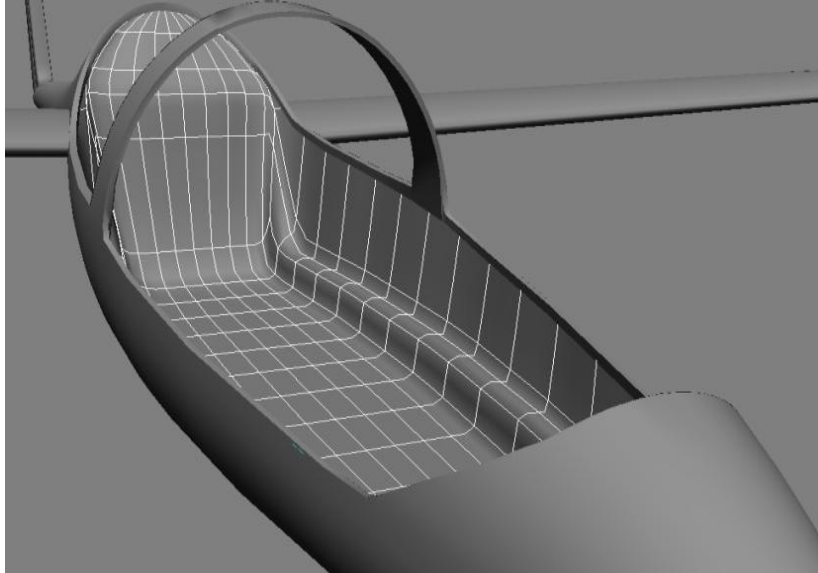
a) wyodrębnianie fragmentu geometrii skrzydła



b) zamykanie wolnych przestrzeni

Rys. 3.15 Proces tworzenia lotek skrzydła

Zastosowanie przezroczystego materiału szyb (więcej w dalszej części rozdziału) powoduje, że aby otrzymać odpowiednią prezencję naszego modelu potrzebujemy wykonać także wypełnienie wnętrza kokpitu, które wykonane zostało jako osobny obiekt. Do tego, nadaliśmy grubość ściankom zewnętrznej części szybowca poprzez odpowiednie wyciągnięcie poligonów na krawędzi okna (Rys. 3.16).



Rys. 3.16 Wypełnienie wnętrza kokpitu

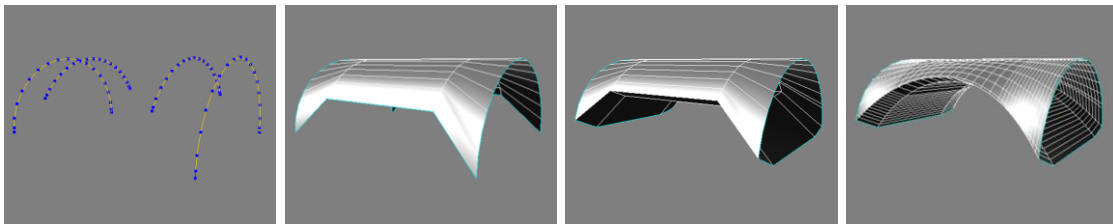
W ten sposób otrzymaliśmy geometrię tworzonego modelu bez uwzględnienia niektórych opcjonalnych detali, które zostaną przedstawione w dalszej części rozdziału. W procesie tworzenia modelu zastosowaliśmy wiele uproszczeń w stosunku do rzeczywistego szybowca. Spowodowane jest to tym, że woleliśmy zrezygnować z niektórych mniej ważnych z wizualnego punktu widzenia elementów szybowca na rzecz poprawienia rozdzielczości elementów, które zostały utworzone.



## Zastosowane narzędzia

- **Loft**

Pozwala tworzyć powierzchnię poprzez łączenie geometrią utworzonych wcześniej krzywych. Utworzoną geometrię możemy kontrolować zarówno za pomocą krzywych jak i wierzchołków powstałej płaszczyzny. Mamy możliwość kontrolowania ilości geometrii jaką wypełniane są przestrzenie między krzywymi poprzez określenie ilości podziałów w kierunkach lokalnych u, v, a także zamykać utworzoną powierzchnie w poszczególnych kierunkach jeśli powstała na bazie krzywych otwartych.



a) krzywe

b) powierzchnia otwarta

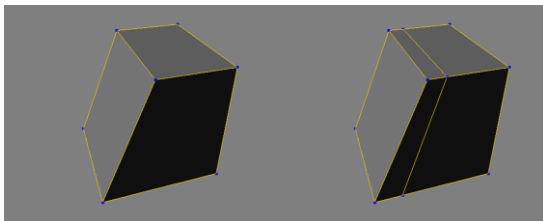
c) powierzchnia zamknięta

d) zwiększony podział

Rys. 3.17 Tworzenie powierzchni przy użyciu funkcji *loft*.

- **Add Edge Loop Tool**

Pozwala dodawać do istniejącego obiektu geometrię w postaci pętli krawędzi. Należy wybrać miejsce na istniejącej krawędzi obiektu. Algorytm następnie sam decyduje które krawędzie należą do tej samej pętli co wybrana krawędź i przeprowadza łamaną podziału tych krawędzi tworząc pętle. Wszystkie krawędzie dzielone są z zachowaniem proporcji długości odcinków powstałych przez podział.



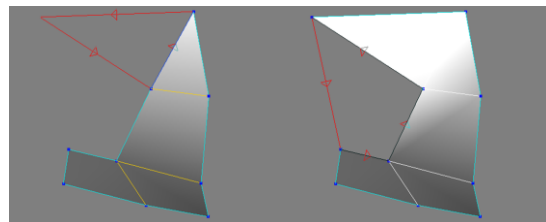
a) obiekt wejściowy

b) dodanie pętli

Rys. 3.18 Zastosowanie *Add Edge Loop Tool*

- **Add/Edit Polygon Tool**

Wszechstronne narzędzie, służące zarówno do tworzenia nowych obiektów, jak i dodawania geometrii do obiektów już istniejących. Działanie tego narzędzia polega na określaniu punktów w przestrzeni gdzie chcemy żeby pojawił się następny wierzchołek tworzonego wielokąta będącego jedną ze ścian obiektu. Poza wyznaczaniem nowych punktów można także wybierać istniejącej już wierzchołki przez co narzędzie pozwala łączyć ze sobą brzegi obiektu.



a) dodanie nowego wierzchołka

b) łączenie krawędzi brzegowych

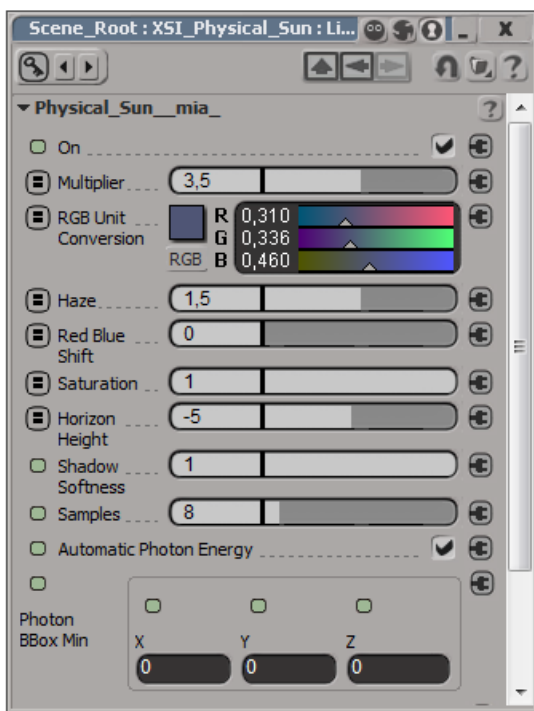
Rys. 3.19 Tworzenie ścian przy użyciu narzędzia *Add/Edit Polygon Tool*

- **Bridge**

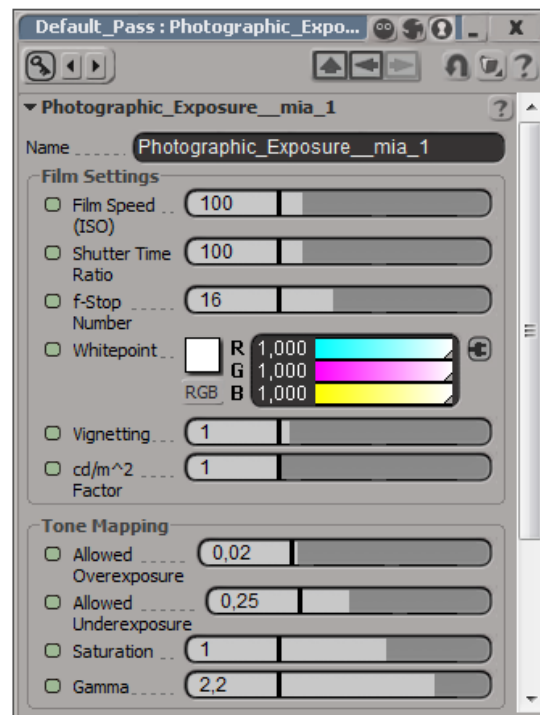
Służy do łączenia geometrią krawędzi brzegowych obiektów otwartych. W odróżnieniu od narzędzia *Add/Edit Polygon Tool* dokonuje tego w sposób automatyczny poprzez określenie maksymalnego zasięgu na jakim mogą znajdować się krawędzie brzegowe, aby zostały ze sobą połączone. Narzędzie to nie daje pełnej kontroli nad sposobem łączenia krawędzi, lecz jego prostota i szybkość działania powodują, że w niektórych przypadkach łączeń jest dobrą alternatywą dla *Add/Edit Polygon Tool*.

### 3.1.2 Oświetlenie i materiały

Zajęliśmy się utworzeniem materiałów dla naszych modeli, dzięki którym będziemy mogli nadać im realistyczny wygląd podczas renderowania. Wygląd konkretnego materiału zależy jednak od oświetlenia sceny, na której znajduje się model. Należy więc najpierw utworzyć odpowiednie oświetlenie. W tym celu posłużyliśmy się wbudowanym w *Softimage* presetem oświetlenia *Physical sun and sky*, który posłuży nam także jako niebo oraz określać będzie położenie słońca na nieboskłonie (ta wiedza przyda się przy tworzeniu modelu fizycznego). *Physical sun and sky* tworzy na scenie oświetlenie typu *infinite* czyli takie, które w danym kierunku generuje światło o takim samym natężeniu niezależnie od odległości oświetlanych obiektów. Zarówno natężenie światła jak i kolor nieba oraz inne opcje jak chociażby położenie wysokości horyzontu ustawić możemy w panelu opcji słońca (Rys. 3.20a). Dodatkowo użycie słońca fizycznego dodaje do sceny ekspozycję fotograficzną, której poddawane są renderowane obrazy. Shader ten konwertuje uzyskany obraz w taki sposób jakby był on widziany okiem kamery. Dzięki temu w łatwy sposób możemy kontrolować parametry końcowego obrazu takie jak chociażby nasycenie i jasność. Następnie wstawiliśmy szybowiec do nowopowstałego otoczenia (Rys. 3.21a,b). Zaobserwować możemy jednak, że od strony cienia model jest bardzo ciemny, ponieważ nie pada z tej strony żadne światło. Zamiast wprowadzać kolejne źródła światła zdecydowaliśmy się włączyć opcję renderowania *Final Gathering* (patrz Final Gathering), co skutkuje realistycznie wyglądającym rozprzestrzenianiem się światła (Rys. 3.21c,d).

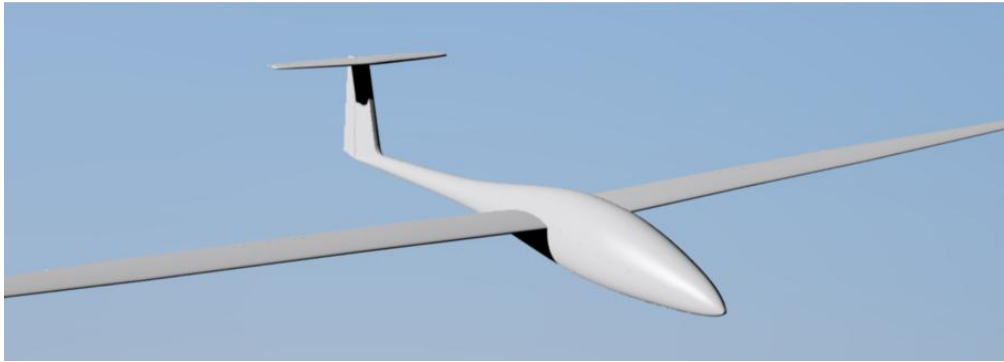


a) opcje oświetlenia i tła

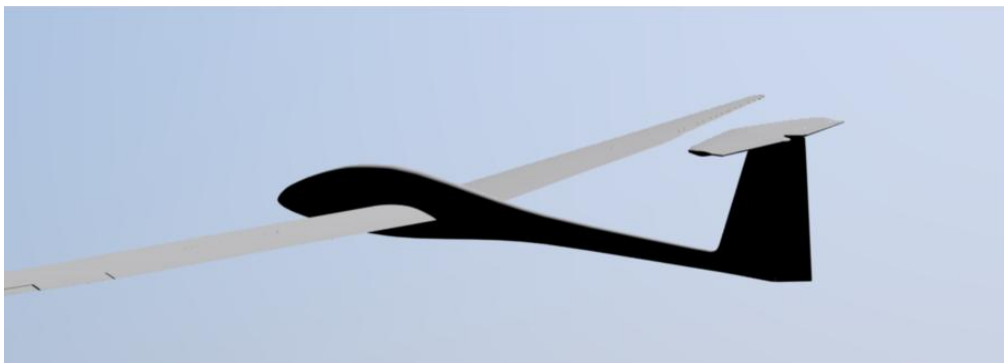


b) opcje ekspozycji fotograficznej

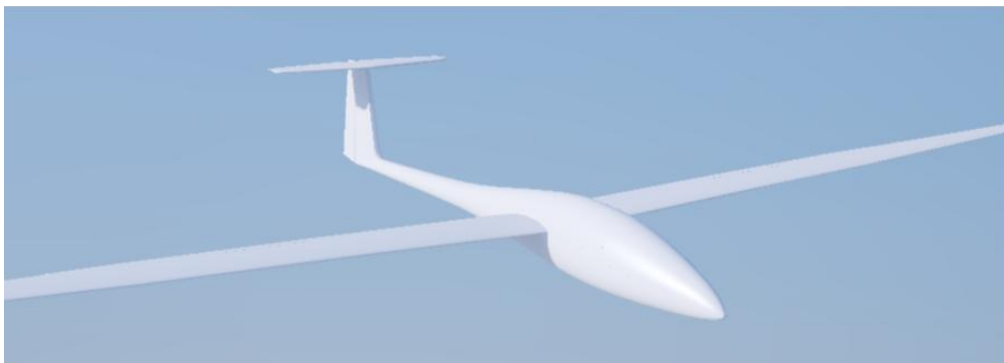
Rys. 3.20 Opcje presetu *Physical sun and sky*



a) wygląd od strony nasłonecznionej



b) wygląd od strony cienia

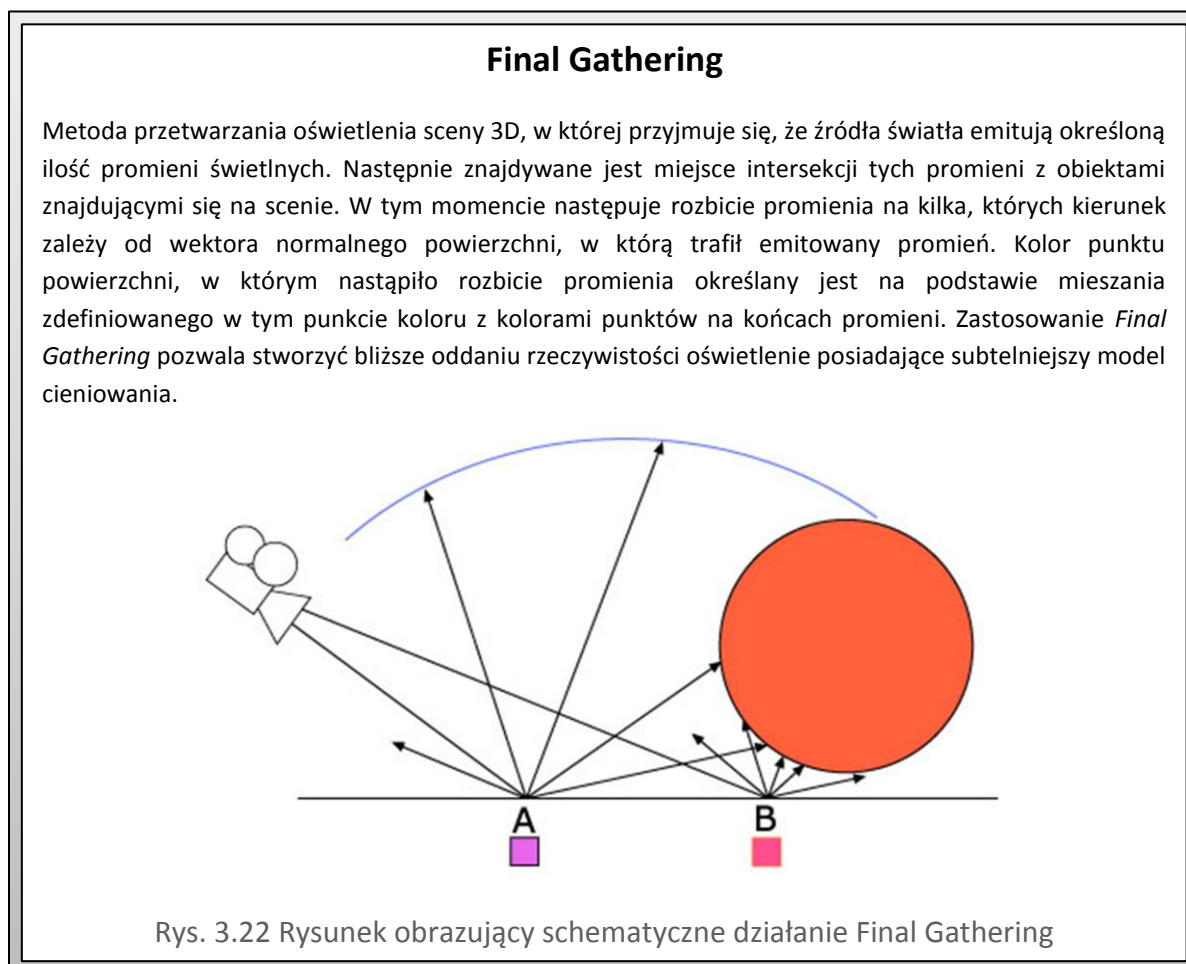
c) wygląd od strony nasłonecznionej po zastosowaniu *Final Gathering*d) wygląd od strony cienia po zastosowaniu *Final Gathering*

Rys. 3.21 Wygląd szybowca w utworzonym środowisku

Następnie przystąpiliśmy do wykonania materiałów dla modelu szybowca. Elementem wspólnym wszystkich utworzonych przez nas materiałów będzie to, że wyprowadzone zostaną z bazowego materiału typu Lambert. Materiał ten posiada możliwość definiowania następujących właściwości:

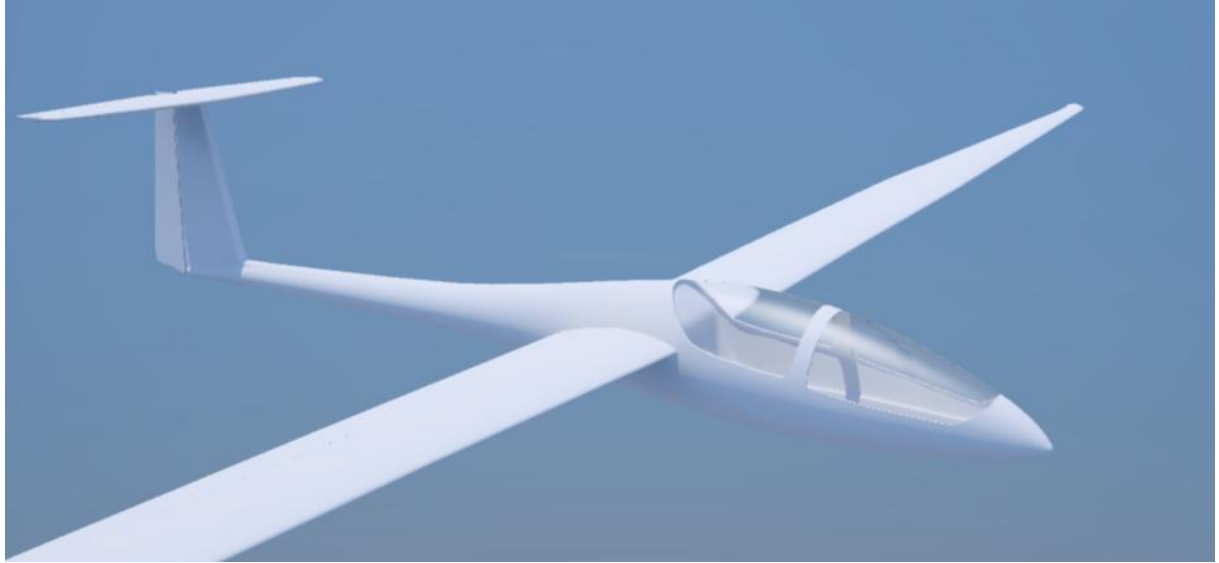
- kolor
- odbłysek – sposób odbijania światła padającego na obiekt
- odbicie – sposób odbijania się w powierzchni obiektów znajdujących się przed nią
- przezroczystość – widoczność obiektów znajdujących się po drugiej stronie powierzchni
- przepuszczalność – ilość światła przepuszczanego przez powierzchnie
- żarzenie – własność, która sprawia, że powierzchnia wydaje się emanować światło o określonym kolorze, które jednak nie rozświetla innych elementów sceny

Każdą z powyższych właściwości można wyłączyć lub włączyć i ustawić jej kolor, który w wielu przypadkach interpretowany jest jako wartość procentowa efektu. Przy wyborze zrezygnowaliśmy z materiałów bardziej złożonych, które mogłyby jeszcze lepiej odwzorować niektóre właściwości (metali, drewna) jednak kosztem zdecydowanie dłuższego czasu renderowania tych elementów. Postanowiliśmy więc utrzymać wszystkie stosowane materiały na podobnym poziomie złożoności.

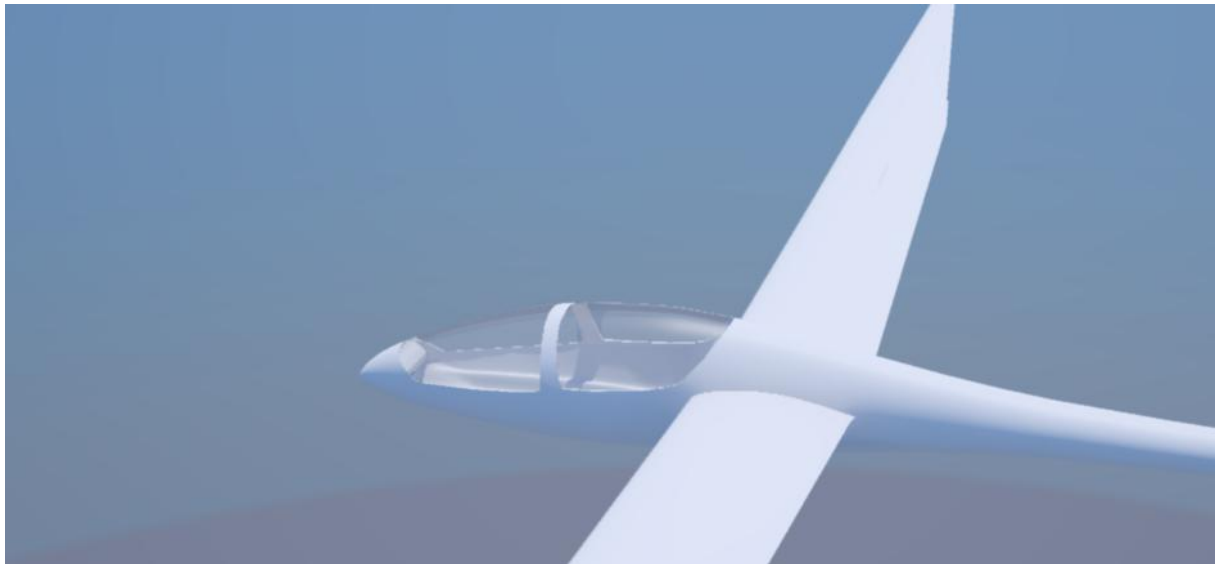


Utworzyliśmy następujące materiały:

- plastik, włókno szklane – cechuje słaby odbłysek światła, kolor dowolny, brak dodatkowych efektów
- szkło – wyłączony kolor, bardzo mocny odbłysek, 100% przezroczystości
- metal - słaby odbłysek światła, słabe odbicie, kolor metaliczny (odcień szarego)
- guma – intensywny odbłysek światła na małym obszarze, kolor dowolny
- drewno (wypolerowane) – średni odbłysek, w miejsce koloru wpięta tekstura proceduralna *wood* (preset środowiska *Softimage*)



a) wygląd od strony nasłonecznionej



b) wygląd od strony cienia

Rys. 3.23 Efekt zastosowania materiałów

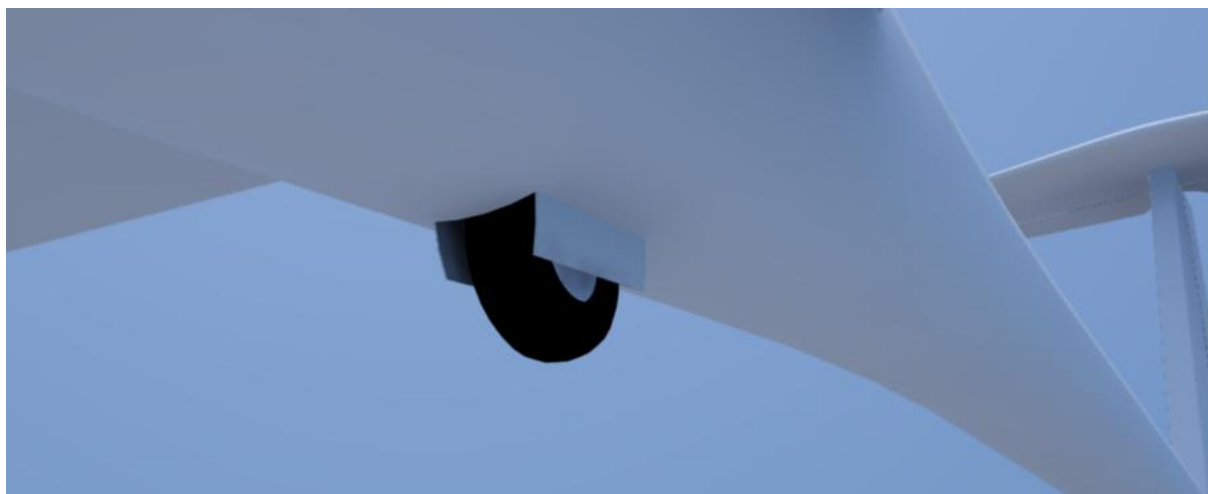
### 3.1.3 Detale

W celu urozmaicenia modelu szybowca wykonaliśmy liczne opcjonalne detale, które w zależności od potrzeb wizualizacji mogą zostać ukryte. Detale wykonane zostały podobnymi technikami jak główne elementy modelu dlatego przedstawimy końcowy efekt wykonania bez wdawania się w szczegóły.

#### 3.1.3.1 Znakowania i koła



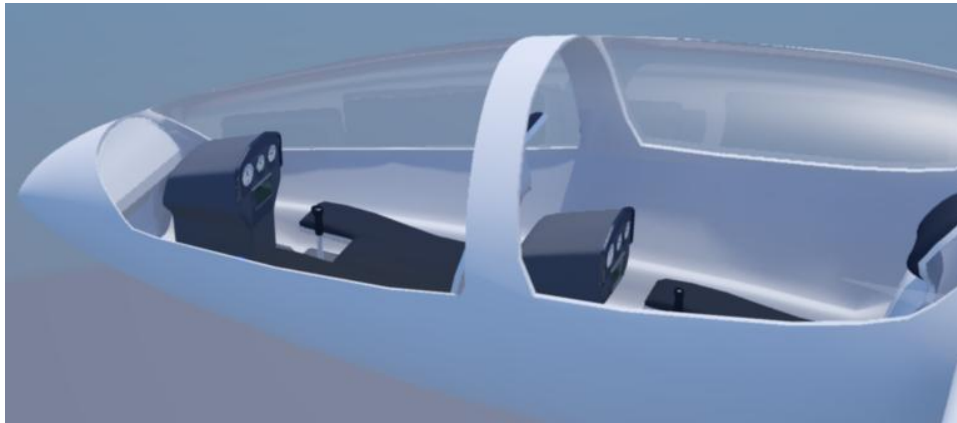
a) „rejestracja” umieszczona jest na ogonie oraz dolnej powierzchni skrzydła



c) Środkowe koło umieszczone jest w zamykanej wnęcie

Rys. 3.24 Koła oraz znakowania – „numery rejestracyjne” i czerwone pasy poprawiające widoczność szybowca

### 3.1.3.2 Wnętrze



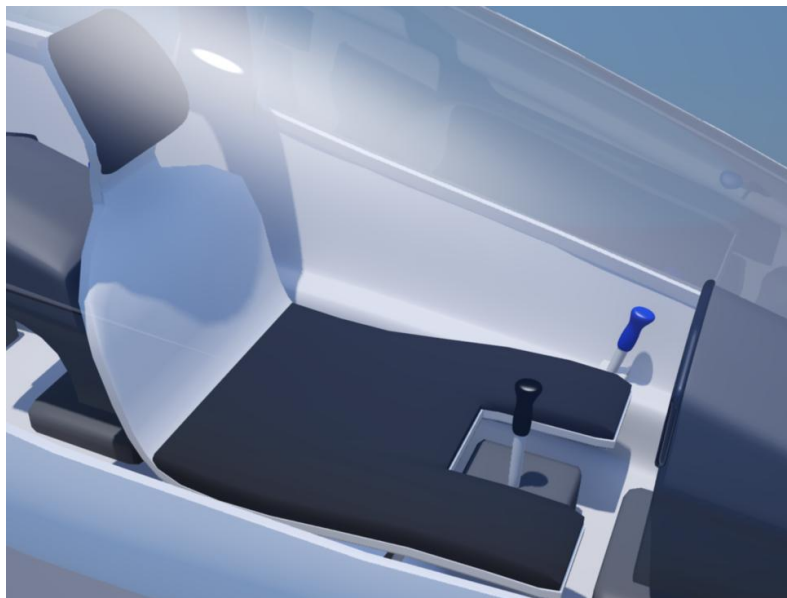
a) widok z boku



b) konsola pasażera



c) konsola kierowcy



c) fotel kierowcy

Rys. 3.25 Wnętrze szybowca

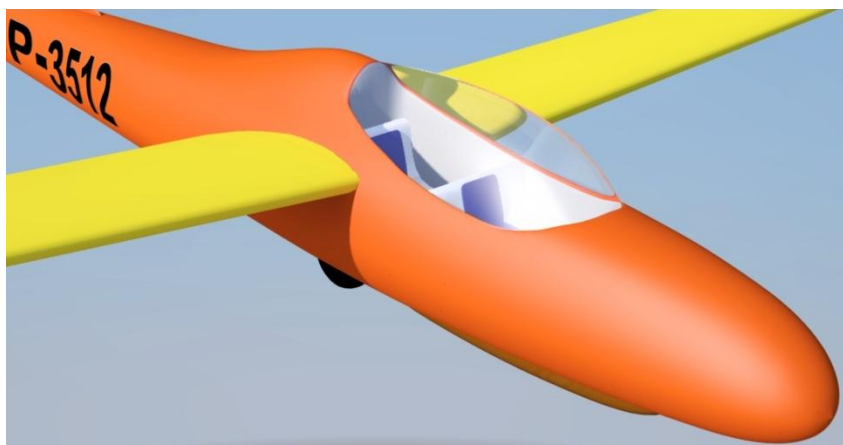
## 3.1.4 Inne modele

### 3.1.4.1 *Schweizer SGS 2-32*

Pierwsze plany konstrukcji tego dwusiedzeniowego szybowca pojawiły się w 1959. Przewidywana wysoka wydajność oraz rozmiary zapewniające komfort podczas lotu przyczyniły się do ochrzczenia go cadillac'iem wśród szybowców firmy Schweizer. Pierwszy lot odbył się trzy lata później w roku 1962. Możliwość zabierania pasażerów oraz wysoka wygoda lotu spowodowała, że szybowiec ten głównie wykorzystywany jest w lotach turystycznych lub ćwiczeniowych. W czasach swojej świetności ustanowił wiele rekordów wysokości oraz odległości lotu szybowcem dwusiedzeniowym. Po dziś dzień pozostaje w użyciu parędziesiąt szybowców tego typu w USA oraz Kanadzie. Nasz model szybowca SGS 2-32 wykonaliśmy opisanymi ówczesnie technikami dlatego też tutaj przedstawimy jedynie wyniki naszej pracy (Rys. 3.26).



a) Powłoka szybowca w całości wykonana była z metalu. Pozwalało to na zastosowanie dowolnej kolorystyki



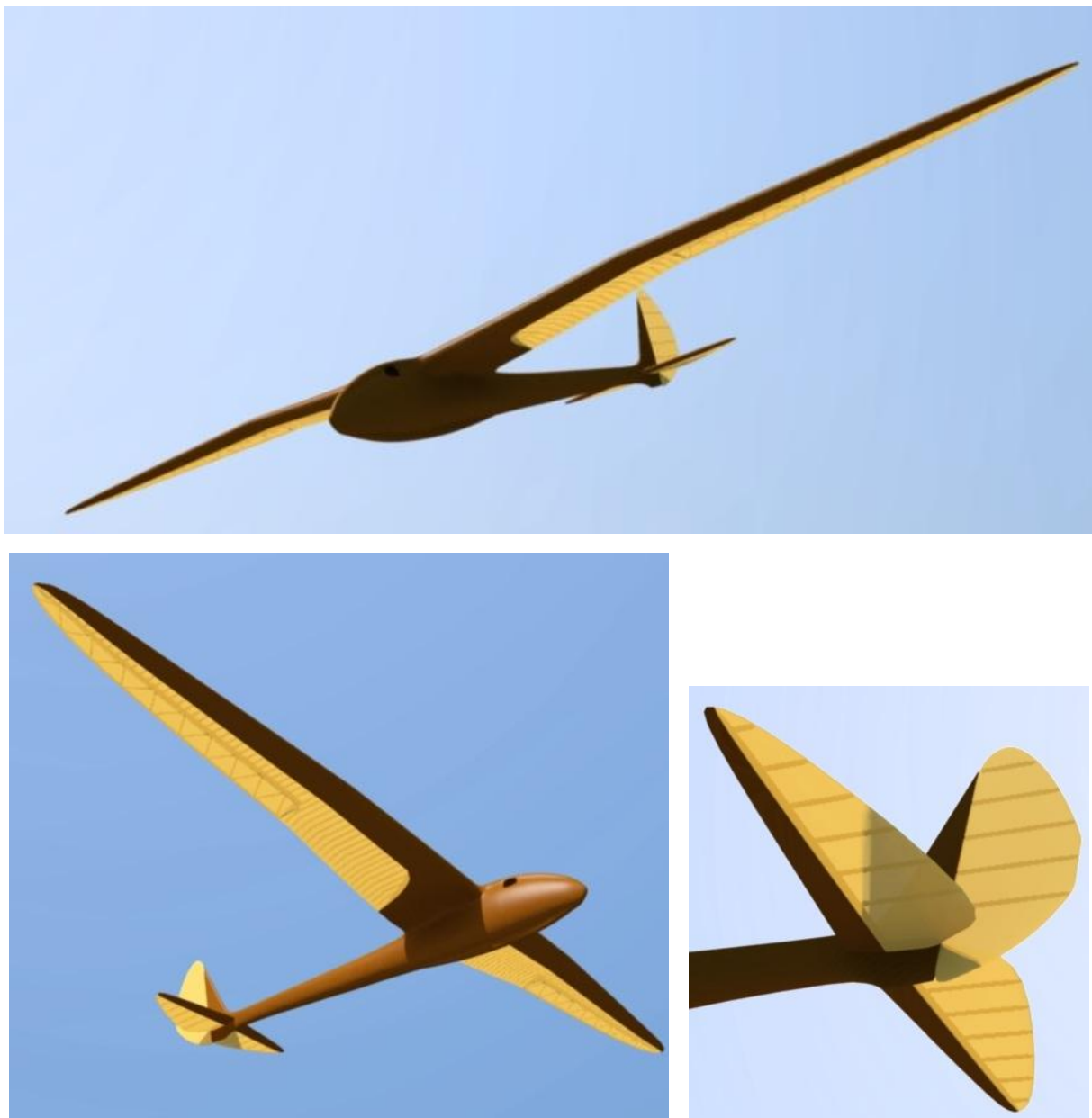
b) Zbliżenie na kokpit

Rys. 3.26 Model Schweizer SGS 2-32



### **3.1.4.2 Fafnir**

Powstanie tego szybowca, który swoją nazwę odziedziczył po mitycznym smoku, sięga lat trzydziestych XX wieku. Konstrukcja szybowca charakteryzowała się bardzo wąskim przekrojem w miejscu kabiny pilota. Sklepienie kokpitu było wykonane w całości z drewna i nie posiadało nawet malutkiej przedniej szyby za to wyposażone było po bokach w dwa iluminatory przez które można było obserwować otoczenie. Skrzydła o rozpiętości 19 metrów szerokie u nasady zwężały się do bardzo wąskich końców i posiadały lekkie skrzywienie mniej więcej w połowie co nadawało im kształt podobny do kształtu skrzydeł mowy, gdy patrzyło się na nie od przodu. Dokładne powody zastosowania takiego kształtu skrzydła nie są znane, ale prawdopodobnie podyktowane było błędnym założeniem, że zwiększa to stabilność w trakcie skręcania. Późniejsze badania wykazały, że skrzydła tego typu wcale nie poprawiają stabilności, a do tego powodują zmniejszenie współczynnika L/D. Skrzydła takie posiadają jednak swoje walory estetyczne i były jednym z powodów, dzięki którym Fafnir okrzyknięty został jednym z najlepiej prezentujących się szybowców w historii. Poza walorami estetycznymi szybowiec ten posiadał także te wydajnościowe dzięki czemu 4-go maja 1931 roku odbył lot na rekordową odległość 278 metrów. Nasz model wykonaliśmy przedstawionymi wcześniej technikami dlatego też przechodzimy od razu do przedstawienia efektu końcowego ( Rys. 3.27 ).

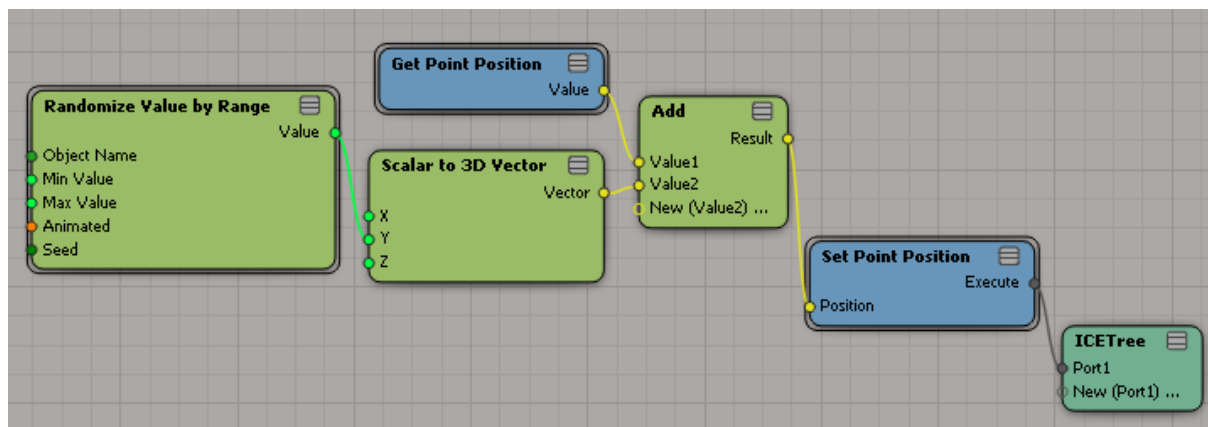


Rys. 3.27 Model szybowca Fafnir

## 3.2 Model otoczenia

W celu stworzenia podłoża, nad którym latać będzie nasz szybowiec posłużymy się środowiskiem *ICE* pakietu *Softimage*. *ICE* (*Interactive Creative Environment*) jest środowiskiem służącym do kontrolowania skomplikowanych systemów graficznych i zarządzania przepływem danych wewnątrz nich. Praca z *ICE* polega przeważnie na wykonaniu trzech kroków:

- Pobranie danych – polega na wprowadzeniu do środowiska danych, które chcemy poddać modyfikacjom. Przykładem danych wczytywanych mogą być położenia punktów modelu, informacje o teksturach i materiale itp.
- Modyfikacja danych – poddanie wczytanych wcześniej danych odpowiednim modyfikacjom jak np. proste operacje arytmetyczne (dodawanie wartości), randomizację, łączenie danych, przekształcanie typu itp.
- Ustawienie danych – załadowanie otrzymanych danych w odpowiednie miejsca tworzonego modelu.



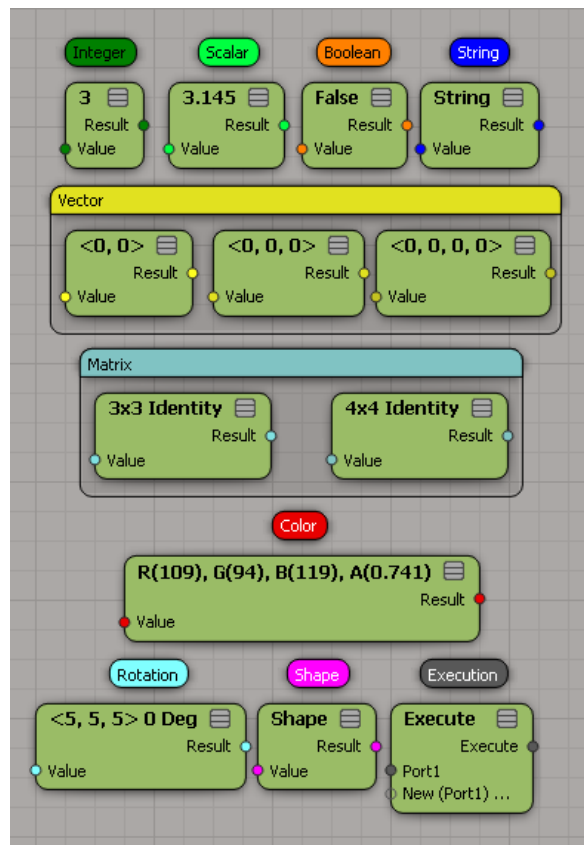
Rys. 3.28 Przykładowe drzewo ICE powodujące losową zmianę położenia wierzchołków obiektu (patrz ICE - przykład).

---

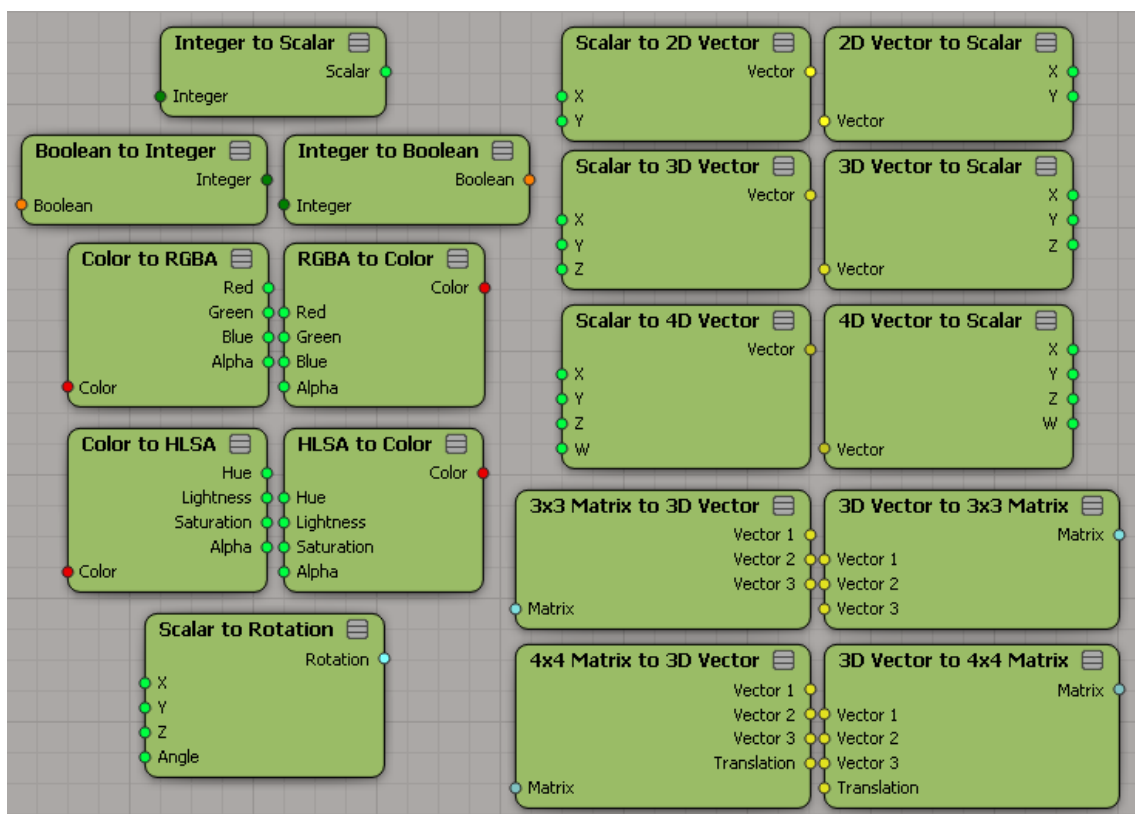
Środowisko *ICE* posiada konstrukcję węzłową (Rys. 3.28). Węzeł posiada porty wejścia oraz wyjścia oraz pewną określoną funkcję. W celu rozpoczęcia korzystania z *ICE* należy w pierwszej kolejności utworzyć węzeł *ICE tree*, który jest korzeniem tworzonego przez nas drzewa. Następnie do drzewa dodajemy odpowiednie węzły i łączymy je między sobą przy pomocy portów wejścia/wyjścia o odpowiadających typach. Każdy port węzła posiada określony typ danych (Rys. 3.29). Każdy typ charakteryzuje inny kolor portów:

- *Integer* – liczby całkowite (kolor portu zielony)
- *Scalar* – wartości zmiennoprzecinkowe (kolor jasnozielony)
- *Boolean* – wartość logiczna (kolor pomarańczowy)
- *Color* – przechowuje dane o kolorze, będące wartościami kanałów RGB oraz kanału *Alpha* przezroczystości koloru (kolor czerwony)
- *Vector* – typ przechowujący dane o położeniu punktu w przestrzeni, wyróżniamy wektory 2D, 3D oraz 4D składające się odpowiednio z dwóch, trzech i czterech wartości skalarnych (kolor żółty o różnym nasyceniu w zależności od typu wektora),
- *Matrix* – Porty danych macierzowych, których wyróżniamy dwa rodzaje - macierze 4x4 oraz 3x3 (kolor ciemnoniebieski)
- *String* – tekstowy typ danych (kolor granatowy)
- *Execution* – nie są tak naprawdę typami danych, służą natomiast do kontroli przepływu danych. Jest to jedyny typ portów jakie posiada korzeń drzewa *ICE tree* (kolor szary)
- *Rotation* – obrót reprezentowany poprzez wektor osi [x, y, z] oraz kąt w stopniach (kolor jasnoniebieski)
- *Shape, Geometry* –opisują odwołanie do obiektów geometrycznych lub kształtów znajdujących się na scenie (kolor fioletowy)
- *Reference* – link do obiektu (kolor ciemnozielony)
- *Polymorphic* – porty czarne są portami polimorficznymi

Środowisko *ICE* posiada ścisłą kontrolę typów co powoduje, że próba łączenia portów o różnych typach nie powiedzie się gdyż *Softimage* zgłosi błąd *Type mismatch*. W celu zmiany pomiędzy poszczególnymi typami dostarczone jest wiele węzłów konwersji typów (Rys. 3.30).



Rys. 3.29 Typy danych środowiska ICE

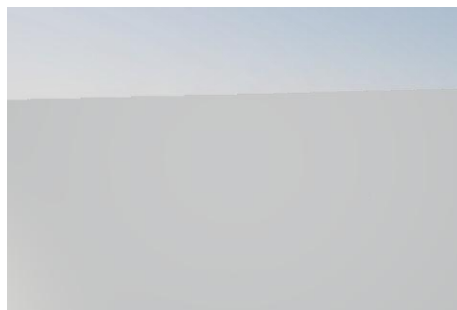


Rys. 3.30 Niektóre węzły konwersji typów ICE

### ICE - przykład

Omawiamy działanie drzewa *ICE* pokazanego na Rys. 3.28. Na drzewie tym posiadamy następujące węzły:

- *Get Point Position* (węzeł pobierający dane) – pobiera położenia wierzchołków obiektu, ponieważ drzewo *ICE* tworzone jest na konkretnym elemencie sceny wystarczy pobierać położenia wierzchołków *Self.PointPosition*
- *Scalar to 3d Vector* (węzeł modyfikujący dane) – przekształca przekazane na wejścia dane skalarne XYZ na wektor 3d o współrzędnych [X, Y, Z]. W naszym przykładzie zmieniamy jedynie wartości składowej Y co spowoduje losowe zmiany wysokości wierzchołków. Reszta portów pozostaje niepołączona co skutkuje inicjalizacją domyślnymi wartościami 0
- *Randomize Value by Range* (polimorficzny węzeł modyfikujący dane) – losuje wartość z przedziału [*Min Value*, *Max Value*] i ustawia ją na wyjście. Porty *Min Value*, *Max Value* oraz *Value* są polimorficzne i poprzez podłączenie wyjścia do wejścia typu *Scalar* zostały zainicjalizowane takim właśnie typem. Wejścia *Min Value* oraz *Max Value* (jak i inne węzły *ICE*) mogą być ustawiane zarówno za pomocą portów wyjściowych innych węzłów jak i poprzez menu kontekstowe konkretnego węzła
- *Add* (polimorficzny węzeł modyfikujący dane) – dodaje do siebie wartości określonego typu przekazane na wejście, w naszym przypadku węzeł zainicjowany został wektorami 3d i na nich operuje
- *Set Point Position* (węzeł ustawiający dane) - ustawia położenia wierzchołków obiektu poprzez odwołanie *Self.PointPosition*, analogicznie jak w przypadku *Get Point Position*
- *Ice Tree* – korzeń drzewa, wykonuje podłączoną do niego sieć węzłów



a) przed zastosowaniem drzewa *ICE*



b) *Min Value* = 0, *Max Value* = 1

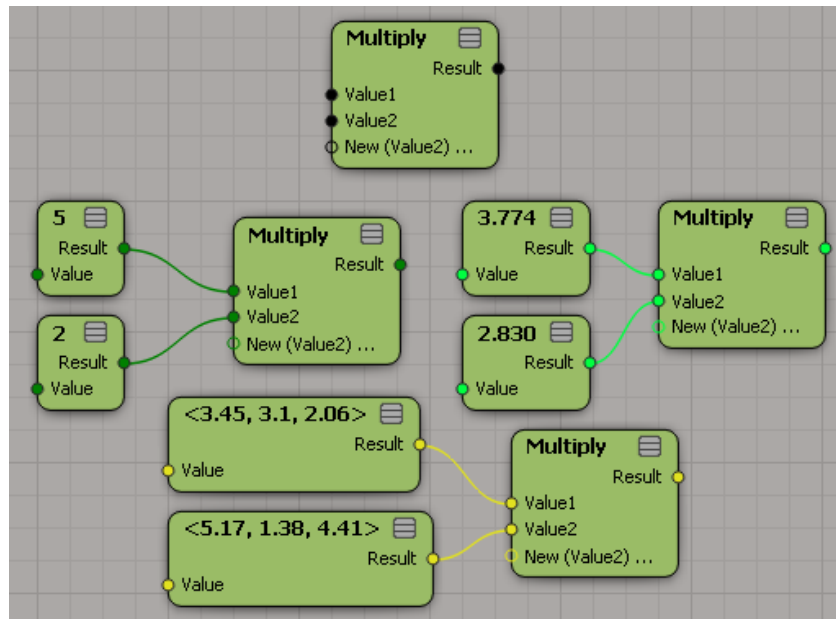


c) *Min Value* = 0, *Max Value* = 2



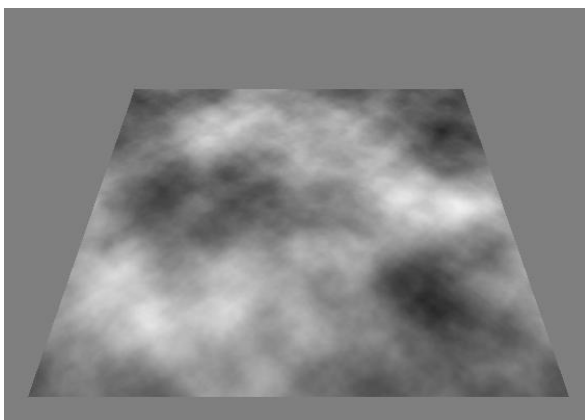
d) *Min Value* = 0, *Max Value* = 3

Rys. 3.31 Efekt działania skonstruowanego drzewa na płaszczyźnie. Po zastosowaniu algorytmu siatka została wygładzona poprzez utworzenie *subdivision surface*. Dla poprawienia wizualizacji zastosowane zostały także odpowiednie ustawienia oświetlenia

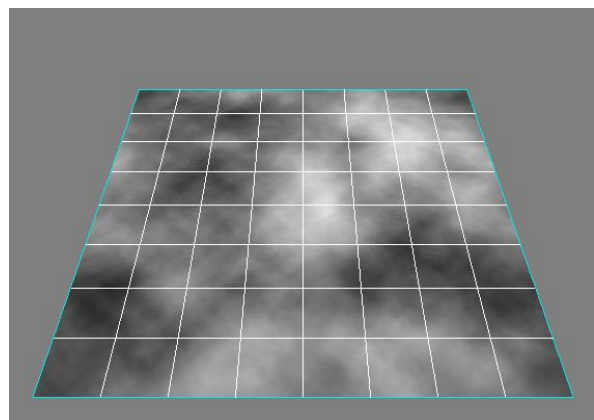


Rys. 3.32 Polimorfizm na przykładzie węzła *Multiply*, który posiada możliwość mnożenia zarówno wartości *Integer*, *Scalar*, *Vector 3D* itp.

*ICE* posiada także węzły polimorficzne (typ danych *Polimorfic*), których typ będzie ustawiany w zależności od podłączanych do nich wartości (Rys. 3.32). Powoduje to zwiększenie uniwersalności węzłów, które mogą wykonywać takie same operacje na różnych typach danych. Porty polimorficzne zazwyczaj mogą przyjmować kilka określonych typów danych, więc posiadają także swoje ograniczenia. Zaobserwować możemy, że po przypisaniu do portu polimorficznego danych o konkretnym typie kolor portu zmienia się. Towarzyszy temu przypisanie portom polimorficznym odpowiedniego typu, co zwane jest inicjacją portów.



a) obiekt wraz z nałożoną teksturą



b) gęstość siatki obiektu

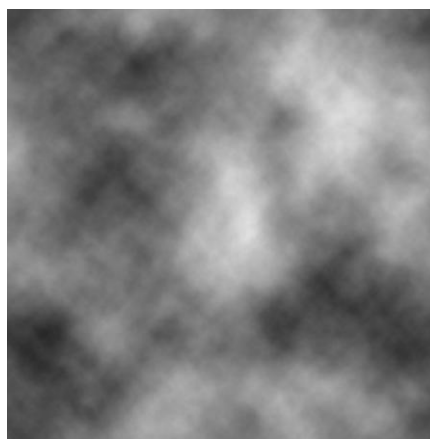
Rys. 3.33 płaszczyzna z teksturą w postaci heightmapy

Algorytm opracowany w systemie *ICE* konstruować będzie geometrię ziemi w oparciu o Heightmapę (patrz **Błąd! Nie można odnaleźć źródła odwołania.**). Tworzymy na

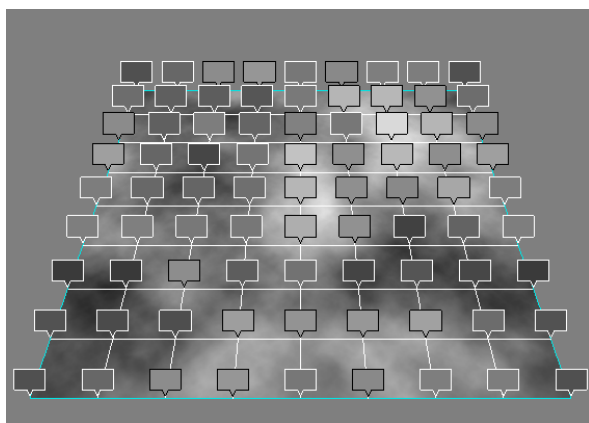
scenie 3D płaszczyzne, a następnie przypisujemy jej teksturę, której obrazem wejściowym jest heightmapa (Rys. 3.33). Posiadając tak określony obiekt możemy utworzyć dla niego drzewo ICE, w którym przy użyciu węzła *Get Point Texture Map* jesteśmy w stanie pobrać wartości koloru tekstury dla każdego piksela (Rys. 3.35a). Następnie przekształcamy reprezentację koloru do postaci HLSA i ekstraktujemy parametr *Lightness* określający jasność koloru w danym punkcie w skali od 0 do 1 (Rys. 3.35b). Posiadając informację o jasności jesteśmy w stanie zmienić położenia punktów w zależności od występującej wartości. W celu zmiany położenia najpierw mnożymy wartość jasności przez parametr kontrolujący rozmiar wypiętrzenia powierzchni, a następnie dodajemy do składowej Y wierzchołka. Uzyskujemy w ten sposób zmianę położenia wierzchołków geometrii w zależności od jasności tekstury im przypisanej (Rys. 3.36). Widzimy, że w zależności od jasności tekstury w danym wierzchołku powierzchnia ulega odpowiedniemu wypiętrzeniu, które możemy kontrolować przy użyciu parametru skali. Geometria na której obecnie działamy posiada jednak zbyt małą rozdzielczość żeby uzyskane rezultaty były satysfakcjonujące. W celu otrzymania dokładniejszych wyników zwiększyliśmy gęstość siatki płaszczyzny (Rys. 3.37). Zaobserwować możemy, że dla siatki 100x100 wypiętrzenie geometrii w dobrym stopniu oddaje dane zawarte w heightmapie. Dla rozdzielczości 600x600 otrzymujemy jeszcze dokładniejszy wynik.

## Heightmapy

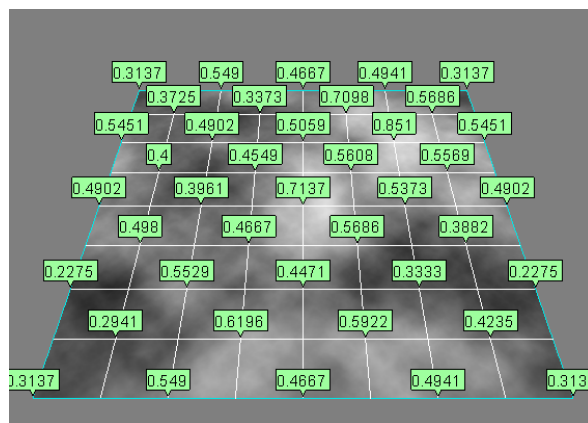
Heightmapą w grafice 3D nazywany jest obraz, w którym przechowywane są dane o ukształtowaniu terenu. W naszym projekcie korzystamy z heightmap w odcieniach szarości, w których poprzez jasność piksela reprezentowana jest wysokość położenia terenu w danym punkcie. Piksele o kolorze czarnym przedstawiać będą najniższe położenie natomiast te o kolorze białym położenie najwyższe.



Rys. 3.34 Przykładowa heightmapa

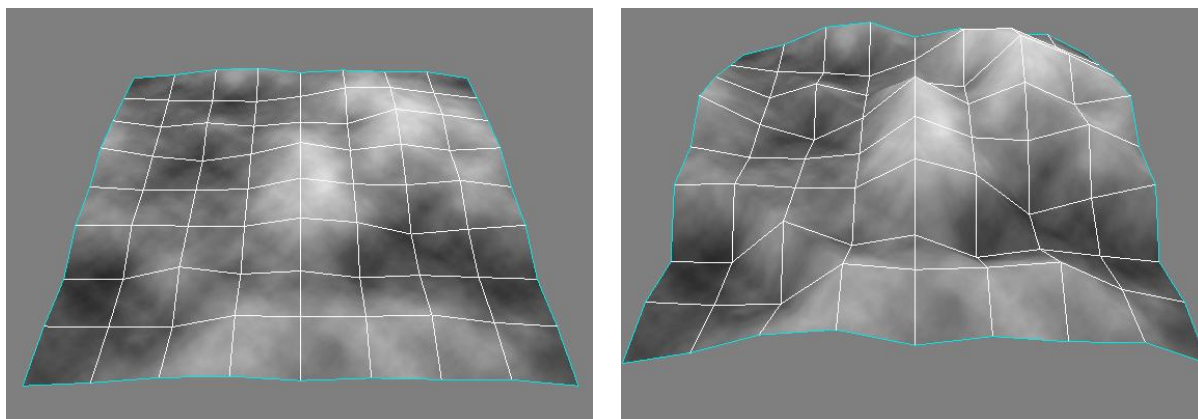


a) wartości kolorów dla poszczególnych wierzchołków



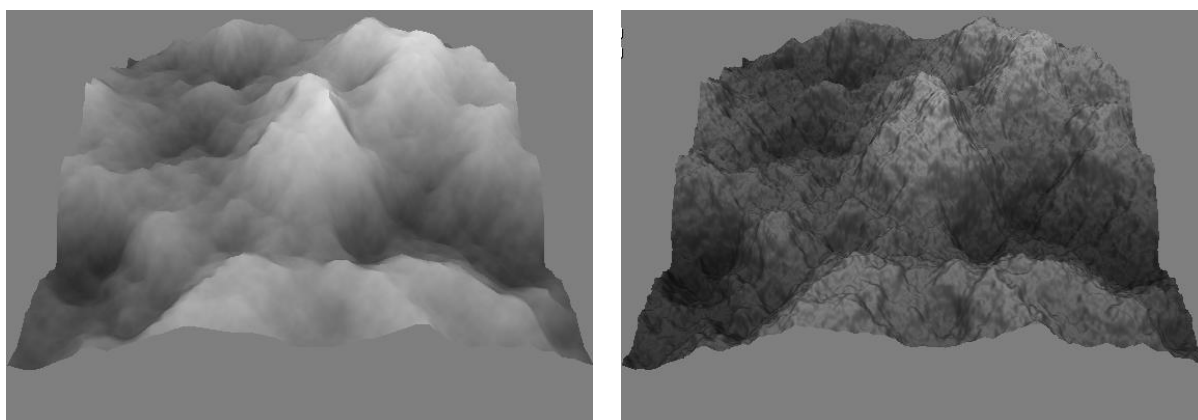
b) wartości parametru *Lightness*



Rys. 3.35 Wartości kolorów oraz odpowiadające im wartości parametru *Lightness*

a) parametr skali wynosi 1

b) parametr skali wynosi 3

Rys. 3.36 Zmiana wysokości położenia wierzchołka w zależności od wartości *Lightness* i parametru skali

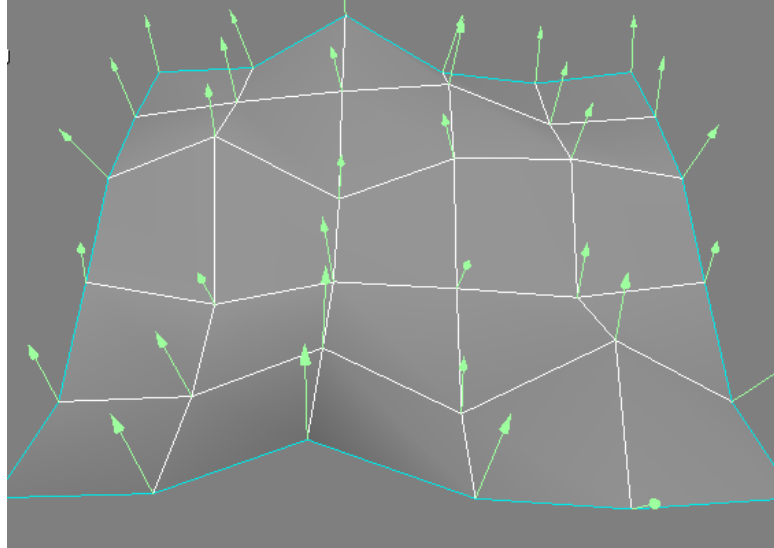
a) siatka 100x100 wierzchołków

b) siatka 600x600 wierzchołków

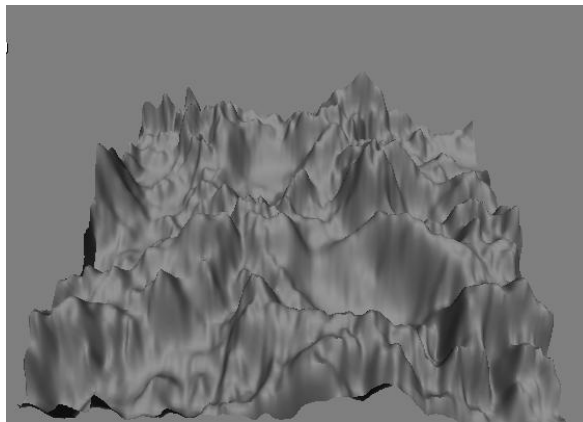
Rys. 3.37 Wygląd generowanego terenu w zależności od gęstości siatki obiektu

Chcemy także posiadać możliwość łatwego teksturowania otrzymanego terenu. W pierwszej kolejności postaramy się wyznaczyć dwie mapy wagowe do wyznaczenia widoczności tekstury reprezentującej skały oraz powierzchnie trawiastą dla wierzchołków geometrii. Założenie jest takie, że im bardziej pochyły fragment powierzchni tym wyraźniejsza będzie składowa tekstury skały w stosunku do tekstury trawy i na odwrót. W tym celu korzystać będziemy z wektorów normalnych powierzchni w węzłach (Rys. 3.38). Zaobserwować możemy, że wierzchołki należące do ścian o większym nachyleniu będą charakteryzowały wektory mniejszej wartości składowej Y wektora normalnego. Skoro tak to do wyznaczenia wartości mapy wagowej tekstury skały weźmiemy długość wektora 2D o składowych  $[X, Z]$  otrzymanych z wejściowego wektora 3D  $[X, Y, Z]$ . Ponieważ wektory normalne są znormalizowane, to długość każdego wynosi 1. Długość wektora  $[X, Z]$  będzie więc należała do przedziału od 0 do 1 i będzie tym większa im mniejsza jest niebrana pod

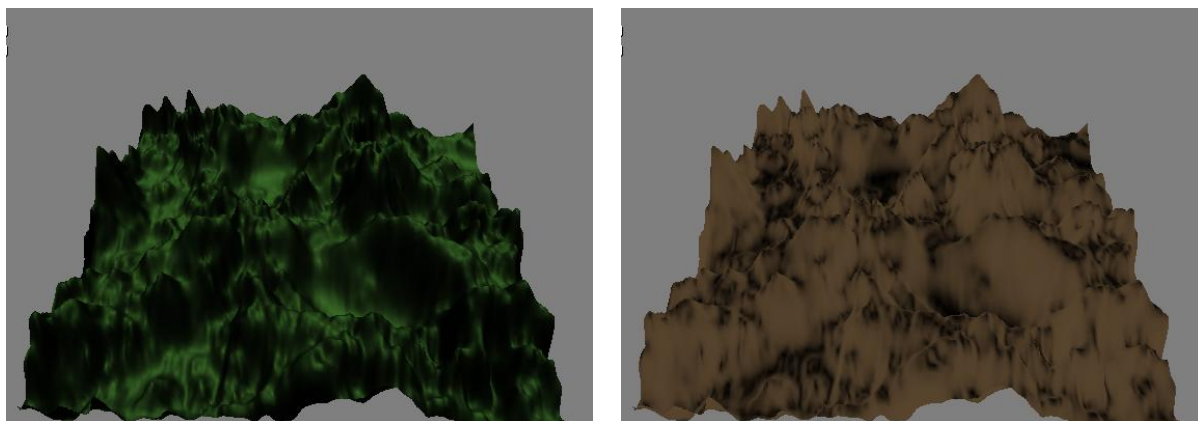
uwagę wartość składowej Y. Dla wektorów w przybliżeniu poziomych, odpowiadających najbardziej stromym fragmentom geometrii będziemy mieć więc wartość równą 1, natomiast dla fragmentów płaskich, w których występuje bardzo silna składowa Y będziemy mieć wartości bliskie 0. Mapę dla obszarów trawiastych otrzymamy poprzez odjęcie od 1 wyniku dla mapy wagowej tekstury skały (Rys. 3.39).



Rys. 3.38 Wektory normalne powierzchni dla poszczególnych wierzchołków



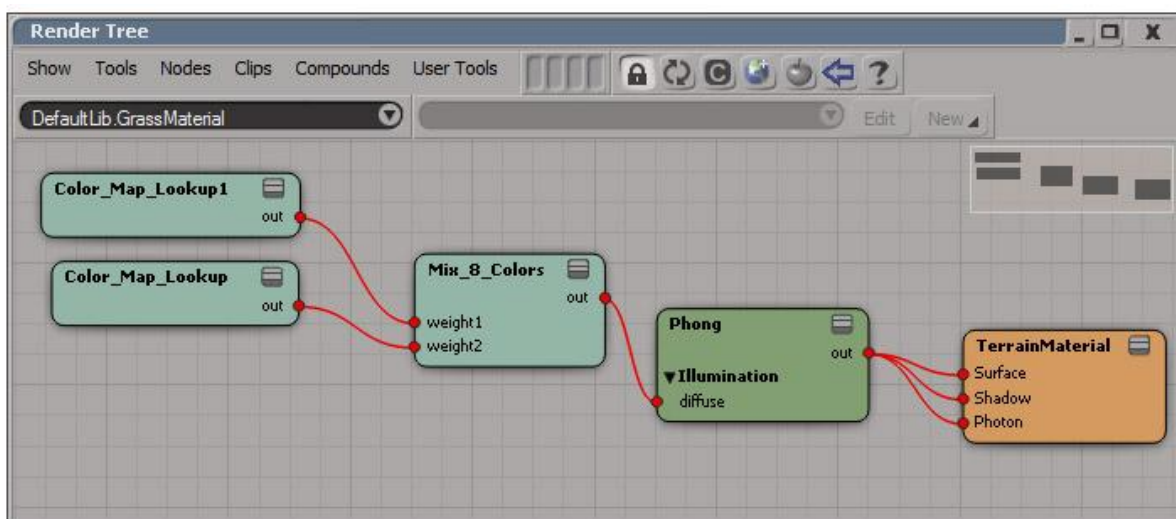
a) geometria wejściowa



b) mapa wag dla tekstury trawy

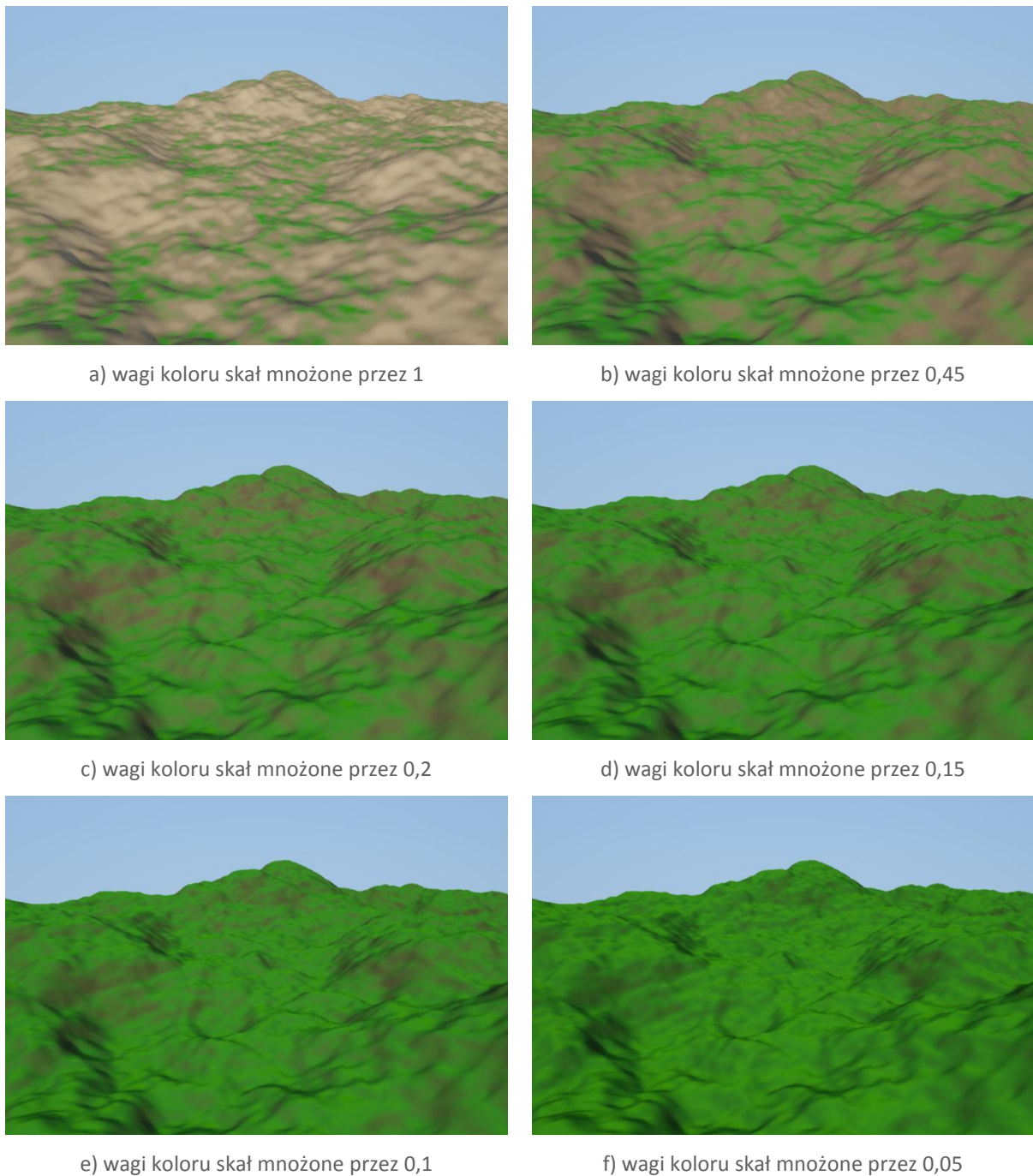
c) mapa wag dla tekstury skał

Rys. 3.39 Tworzenie map wagowych w zależności od wektorów normalnych powierzchni



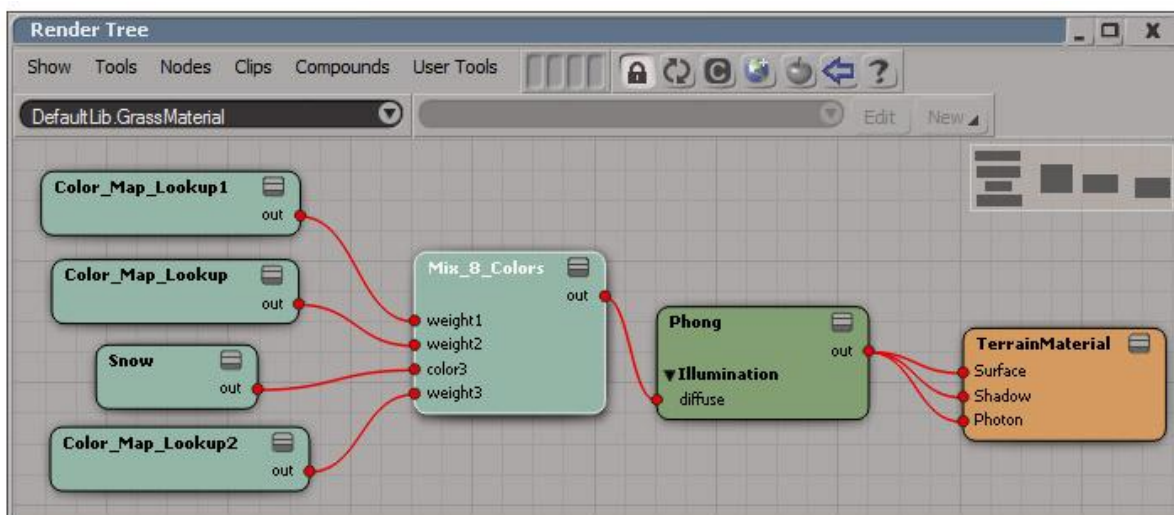
Rys. 3.40 Materiał przypisany stworzonemu terenowi

Następnie utworzyliśmy materiał dla stworzonej przez nas geometrii (Rys. 3.40). W tym celu skorzystaliśmy z węzła *Mix\_8\_Colors*, który pozwala mieszać ze sobą do ośmiu kolorów wejściowych, przy czym przez kolor wejściowy rozumieć należy wszystko co można przyłączyć do portu *diffuse* materiału, czyli zarówno jednolity kolor jak i teksturę, a także węzeł lub sieć węzłów generujących odpowiedni kolor materiału. W naszym przypadku ustawiliśmy kolory jednolite. Korzystamy z węzła *Mix\_8\_Colors*, ponieważ pozwala on ustawiać mapy wagowe dla mieszanych kolorów. W związku z tym w odpowiednie porty włączyliśmy uzyskane wcześniej mapy wagowe dla skał i trawy w ten sposób uzyskując materiał kolorujący geometrię z odpowiednimi wagami. Po wyrenderowaniu uzyskanego obiektu zaobserwować możemy (Rys. 3.41a), że tereny skały są zdecydowanie zbyt rozległe i narzucające się. Na szczęście poprzez użycie *Mix\_8\_Colors* możemy w łatwy sposób zmniejszyć intensywność koloru skał w stosunku do koloru trawy zmniejszając ich współczynnik *Alpha* z zaznaczeniem wewnątrz węzła, że zmiana tego kanału ma mieć wpływ na wagi uzyskane z mapy wagowej (opcja *Multiply Weight by Alpha* węzła *Mix\_8\_Colors*).



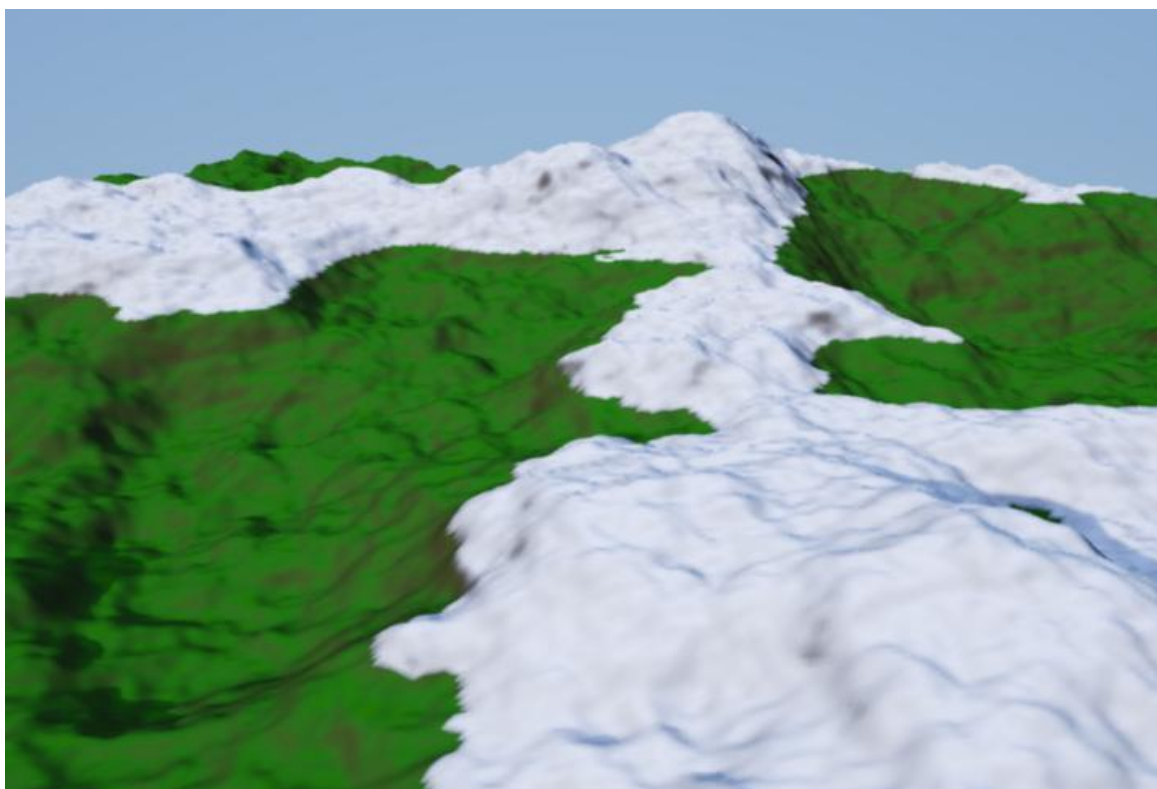
Rys. 3.41 Wpływ mnożnika wag koloru skał na wygląd terenu

Postanowiliśmy do naszego generatora terenu wprowadzić także kolor śniegu dla najwyższych elementów generowanego terenu. W tym celu wyłapujemy wierzchołki, które posiadają wysokość większą niż określona w dodanym parametrze granicznej wysokości trawy, którego wartość można dowolnie ustawiać. Dodatkowo dodajemy do węzła *Mix\_8\_Colors* nowy kolor reprezentujący śnieg. Tym razem jednak zamiast zwykłego koloru jednorodnego podajemy na wejście węzeł *Snow* (Rys. 3.42), wewnątrz którego w łatwy sposób, możemy ustawić parametry reprezentacji śniegu jak choćby jego grubość.

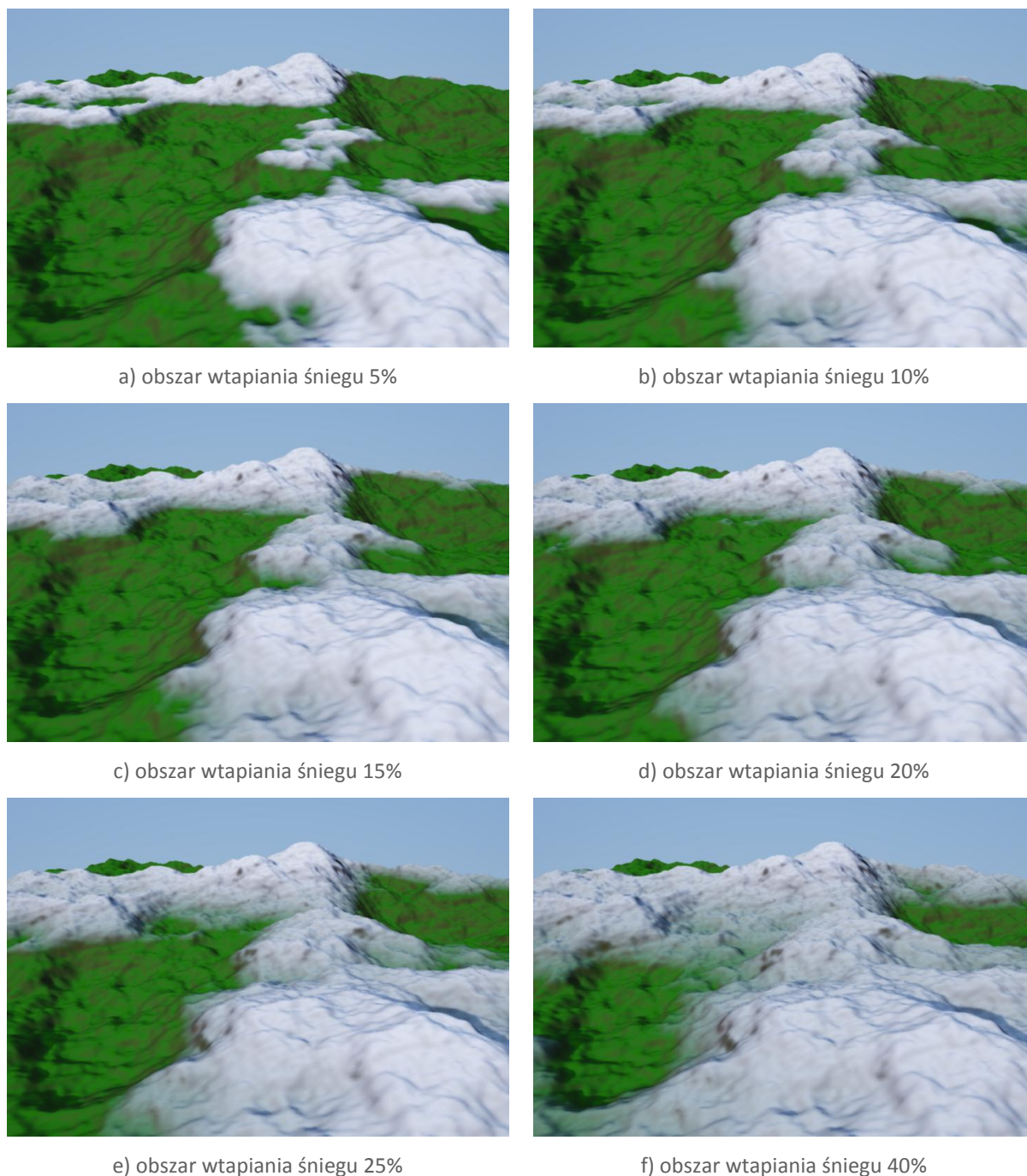


Rys. 3.42 Materiał z uwzględnieniem koloru śniegu

Takie ustawienie generatora koloru śniegu powoduje jednak, że występuje bardzo wyraźna granica pomiędzy trawą, a śniegiem, co wygląda nienaturalnie, a do tego powoduje pojawienie się wyraźnych pikseli w regionach zmiany koloru (Rys. 3.43). W celu wyeliminowania tego nienaturalnego efektu wprowadziliśmy parametr nazwany wtapieniem śniegu, który powoduje, że na określonym przedziale wysokości wokół granicy zmiany koloru następuje liniowe przejście wraz ze wzrostem wysokości od wartości 0 do 1 wagi koloru śniegu. Parametr ten zadawany jest w procentach wysokości punktu granicznego.

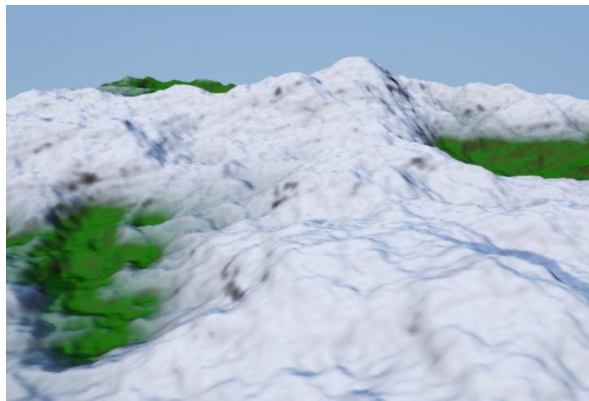


Rys. 3.43 Teren ze śniegiem

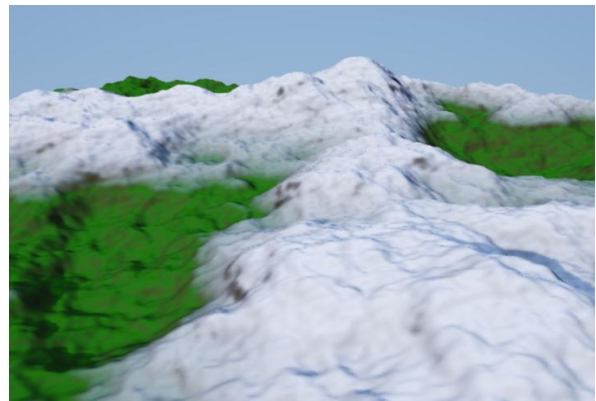


Rys. 3.44 Zmiana parametru wtapiania śniegu

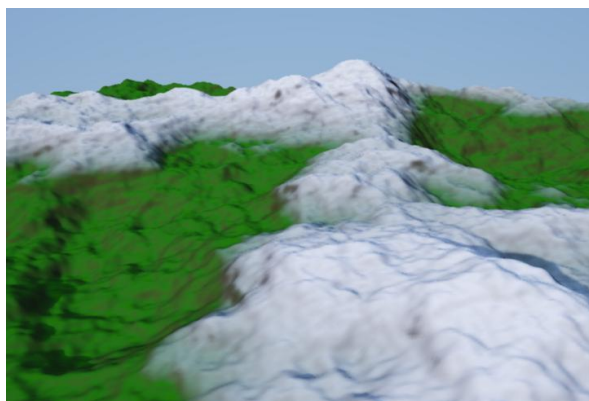
Widzimy (Rys. 3.44), że wprowadzenie tego parametru zdecydowanie poprawia wiarygodność uzyskanego terenu poprzez wygładzenie przejścia między kolorami śniegu oraz trawy. Należy zwrócić uwagę, że zwiększenie wartości parametru poza zwiększeniem łagodności przejścia między kolorami powoduje także zwiększenie zakresu występowania śniegu w niższych partiach terenu. Na szczęście posiadamy także parametr odpowiedzialny za wysokość granicy pokrywy śnieżnej poprzez manipulację którym możemy ustawić śnieg na interesującej nas wysokości (Rys. 3.45).



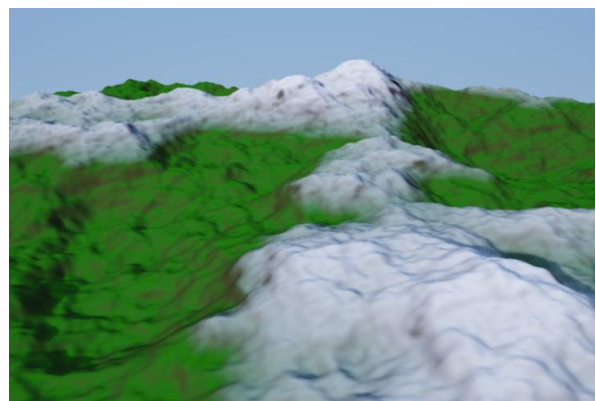
a) wysokość śniegu 1,2



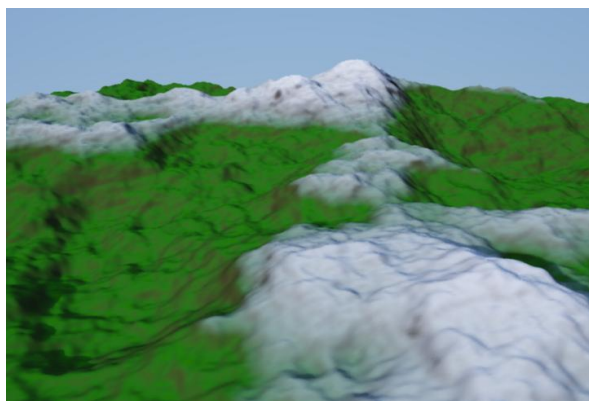
b) wysokość śniegu 1,5



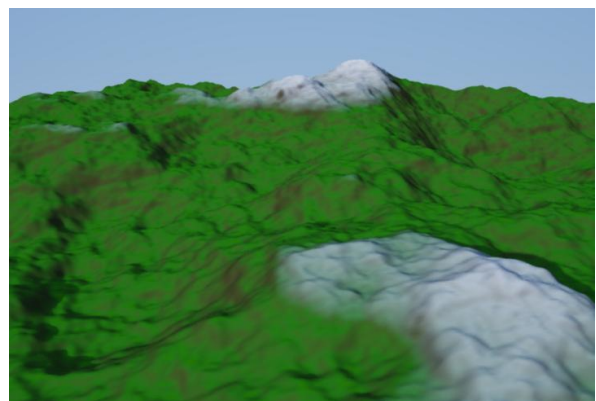
c) wysokość śniegu 1,75



d) wysokość śniegu 1,9



e) wysokość śniegu 2



f) wysokość śniegu 2,3

Rys. 3.45 Zmiana wyglądu terenu w zależności od wysokości pokrywy śniegu, dla parametru wtapiania 20%

# 4 Symulacja

Jak już wspomnieliśmy przy samym wstępie pracy, naszym celem zarówno przy budowaniu modelu graficznego jak i przy pisaniu symulacji było możliwie najdokładniejsze odwzorowanie rzeczywistości. Dodatkowymi celami, jakie sobie postawiliśmy było umożliwienie sterowania szybowcem w czasie rzeczywistym. Zależało nam na możliwości sterowania lotkami, klapami, hamulcami aerodynamicznymi, sterem kierunku oraz wysokości. Szybowiec miał reagować na te czynniki zgodnie z działającymi siłami i ich momentami.

## 4.1 Model fizyczny

W naszej pracy, ponieważ nie projektowaliśmy nowego modelu szybowca od podstaw, jedynie staraliśmy się wiernie odwzorować istniejące modele, mieliśmy komfort posiadania gotowych danych dotyczących konkretnych profilów skrzydeł. Stąd też mogliśmy otrzymać realistyczne wyniki symulacji w sposób o wiele wydajniejszy niż poprzez numeryczne rozwiązywanie równań Naviera - Stokesa. Wszystkie występujące i uwzględniane przez nas zjawiska fizyczne zostały opisane w części teoretycznej, stąd tutaj zajmujemy się jedynie przytoczeniem tych z nich, które wykorzystamy przy symulacji oraz podaniem wykorzystywanych wzorów.

Naturalnym wyborem było potraktowanie szybowca jako trójwymiarowej bryły sztywnej o określonych momentach bezwładności, środku masy i punktach przyłożenia sił. Zajmować się więc będziemy dwoma typami ruchu – postępowym, wyliczonym dla środka ciężkości szybowca oraz ruchem obrotowym, dokonywanym wokół trzech osi.

### 4.1.1 Obliczenia dotyczące sił i ruchu – metoda Eulera, Tensory i Kwaterniony.

Zacniemy od sposobu wyliczenia przemieszczenia naszego szybowca. Dla wyznaczonego wcześniej środka ciężkości, po zsumowaniu wszystkich sił otrzymamy wektor siły wypadkowej  $F_w$ .



### RUCH LINIOWY – METODA EULERA

Aby wyliczyć przemieszczenie dla kroku czasowego  $dt$  posłużyliśmy się metodą Eulera. Wymaga ona użycia rozwinięcia funkcji w szereg Taylora. Jeśli za funkcję przyjmiemy prędkość  $v$ , możemy zapisać:

$$v(t + dt) = v(t) + (dt)v'(t) + \left[ \frac{(dt)^2}{2!} \right] v''(t) + \dots$$

Równanie 15 Rozwinięcie prędkości w szereg Taylora

Ponieważ wystarczy nam tutaj dokładność pierwszego rzędu, dla czasu  $dt=1/24[s]$  (24 klatki na sekundę), a dodatkowo nie mamy żadnych danych dotyczących drugiej pochodnej prędkości  $v''(t)$ . Otrzymujemy w ten sposób wzór:

$$v(t + dt) \approx v(t) + (dt)v'(t)$$

Równanie 16 Rozwinięcie prędkości – dokładność pierwszego rzędu

Jako  $v'$  w każdym kroku czasowym pojawi się przyspieszenie  $a=F_w/m$ , gdzie  $m$  to całkowita masa szybowca.

### RUCH OBROTOWY

Przy zajmowaniu się zagadnieniem ruchu obrotowego w 3D należy pamiętać o konieczności zastosowania odpowiednich obiektów matematycznych takich jak tensory momentu bezwładności a także kwaterniony, których to zdecydowaliśmy się użyć ze względu na korzyści jakie dają podczas programowania. Najpierw przypomniane zostaną podstawowe zagadnienia związane z ruchem obrotowym bryły sztywnej.

Moment siły  $\mathbf{M}$  można potraktować jako odpowiednik siły, tak samo jak kilka innych wielkości: momentu pędu  $\mathbf{L}$  – odpowiednika pędu, tensor momentu bezwładności  $\mathbf{I}$  – zamiast masy, prędkość kątową  $\boldsymbol{\omega}$  a także przyspieszenie kątowe  $\boldsymbol{\varepsilon}$ . I tak:

$$\mathbf{M} = \mathbf{r} \times \mathbf{F}$$

Równanie 17 Moment siły

$$\mathbf{M} = \mathbf{I}\boldsymbol{\varepsilon} = \mathbf{I}\left(\frac{d\boldsymbol{\omega}}{dt} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}\right)$$

Równanie 18 Moment siły jako iloczyn momentu bezwładności i zmiany prędkości w czasie

## TENSOR MOMENTU BEZWŁADNOŚCI

Tensor to obiekt matematyczny będący uogólnieniem wektora. Tensorem zerowego rzędu jest skalar, tensorem pierwszego rzędu jest wektor, a tensorem drugiego rzędu macierz 3x3. Jest to bardzo szerokie pojęcie, ale w tym przypadku interesuje nas jedynie możliwość zastosowania go do opisanie momentu bezwładności  $I$  – jako anizotropowej wielkości fizycznej. Dla szybowca moment bezwładności dla każdej z osi obrotu będzie przyjmował inną wartość. Zaletą zastosowania tensora jest konieczność wyliczenia jego oraz jego odwrotności tylko raz, na początku symulacji, i używanie później tych gotowych macierzy przy każdym obliczaniu kąta obrotu szybowca, zgodnie z Równanie 18.

Wspomniany tensor przyjmuje następującą postać:

$$\mathbf{I} = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{xy} & I_{yy} & -I_{zy} \\ -I_{xz} & -I_{zy} & I_{zz} \end{bmatrix}$$

Gdzie każdy ze składowych momentów bezwładności obliczany jest z odpowiednich równań.

## OBROT CIAŁA SZTYWNEGO

Obrót ciała sztywnego w dwóch wymiarach posiada jeden stopień swobody i do jego opisu wystarczy pojedynczy skalar. W trzech wymiarach stopnie swobody są trzy, a więc można by ograniczyć się do podania jedynie trzech kątów Eulera, jak na Rys. 4.1. Takie rozwiązanie w symulacjach przysparza jednak kilku komplikacji – jak chociażby nieokreśloność dwóch kątów, gdy trzeci z nich przyjmuje wartość  $\pi/2$ , czy też problem z dzieleniem przez zero, wynikający ze sposobu zapisu równań ruchu przy użyciu kątów Eulera.

Ulepszeniem tej metody jest zastosowanie macierzy obrotu 3x3 w taki sposób, aby współrzędne wektora  $w'$ , czyli wektora  $w$  po obrocie dało się przedstawić prostym równaniem, jak Równanie 19.

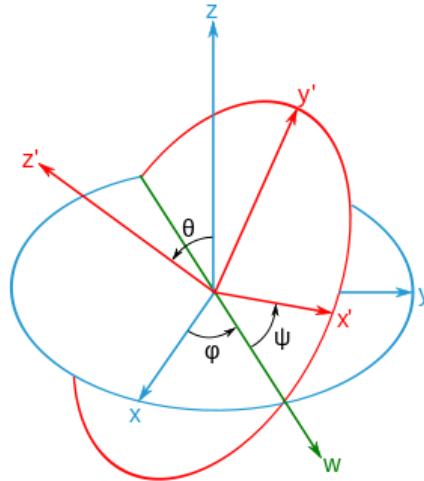
$$w' = R w$$

Równanie 19 Obrót wektora przy pomocy macierzy obrotu

W rzeczywistości, każdy obrót w 3D wyliczamy w ogólności jako złożenie trzech obrotów – wokół osi  $O_x$ ,  $O_y$  i  $O_z$ . Każdemu z nich odpowiada macierz  $R_x$ ,  $R_y$  i  $R_z$ . Dzięki takiemu podejściu orientację szybowca można by zapisywać w każdym kroku po prostu jako wspomnianą macierz obrotu, będącą iloczynem  $R=R_x R_y R_z$ . Po początkowym ustaleniu macierzy wypadkowej co krok programu (klatkę symulacji) wystarczyło by ją aktualizować

według następującej zależności:  $\frac{dR}{dt} = \Omega R$ , gdzie  $\Omega = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$ . Problemem jest

tu jednak konieczność przechowywania i modyfikowania aż 9-ciu zmiennych, ponadto za każdym razem wymagane jest sprawdzenie warunku ortogonalności macierzy  $\mathbf{R}$  oraz zapewnienie warunku  $\det \mathbf{R} = 1$ . Problem pojawia się przy wielokrotnych obliczeniach numerycznych, chociażby z powodu błędu obcięcia. Wtedy to macierz  $\mathbf{R}$  zaczyna również skalować i przesuwać obracany obiekt.



Rys. 4.1 Kąty Eulera

## KWATERNIONY

Rozwiązaniem okazuje się zastosowanie kwaternionów – matematycznych obiektów wprowadzonych, w połowie XIX wieku przez irlandzkiego matematyka Williama Hamiltona, jako rozwinięcie pojęcia liczb zespolonych. Obecnie wykorzystywane są najczęściej w grafice komputerowej właśnie przy zagadnieniu obrotu brył. Ogólna postać kwaternionu  $\mathbf{q}$ , to:

$$\mathbf{q} = q_0 + q_x \mathbf{i} + q_y \mathbf{j} + q_z \mathbf{k}$$

co można także zapisać w postaci

$$\mathbf{q} = q_0 + \mathbf{v}, \quad \text{gdzie } \mathbf{v} \text{ to wektor } \mathbf{v} = q_x \mathbf{i} + q_y \mathbf{j} + q_z \mathbf{k}$$

Dla obrotu, wektor  $\mathbf{v}$  wskazuje oś obrotu, a kwaternion obrotu o kąt  $\theta$  wokół osi wyznaczonej przez wektor  $\mathbf{v}$  zapiszemy następująco:

$$\mathbf{q} = \left[ \cos \frac{\theta}{2}, \sin \frac{\theta}{2} \mathbf{v} \right]$$

Po utworzeniu kwaternionu dla początkowej orientacji modelu, aktualizacja obrotu odbywa się przez zastosowanie poniższego Równanie 20:

$$\frac{d\mathbf{q}}{dt} = \frac{1}{2} \boldsymbol{\omega} \mathbf{q}$$

Równanie 20 Aktualizacja kwaternionu obrotu

Gdzie wektor prędkości kątowej  $\boldsymbol{\omega}$  zamieniony jest na kwaternion  $[0, \boldsymbol{\omega}]$ .

## 4.1.2 Schemat programu

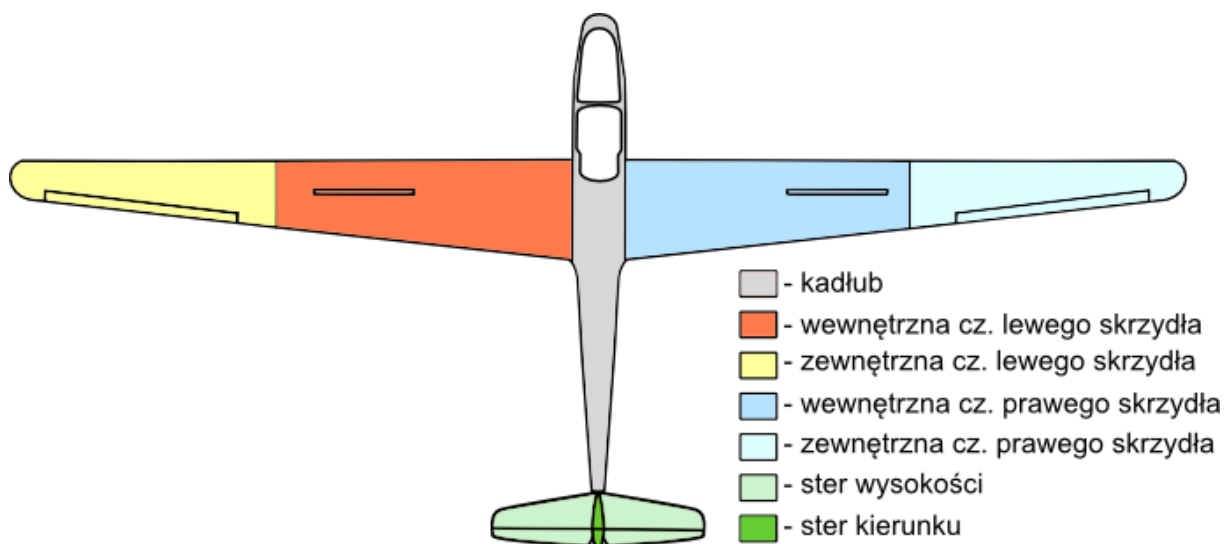
Do obliczenia potrzebnych sił, nasz program wykonuje pięć kroków. Tutaj opisane zostaną skrótowo aby dać pogląd działania programu, a bliżej przyjrzymy się im w części 0, gdzie znając już nieco opis środowiska programistycznego z następnego rozdziału 4.2 będziemy się odwoływać do konkretnych funkcji zastosowanych w programie.

### 1. PODZIAŁ SZYBOWCA NA ELEMENTY.

Podziału dokonaliśmy ze względu na różnice w funkcjach i cechach poszczególnych elementów. Tak więc szybowiec został podzielony na 7 elementów (Rys. 4.2) – skrzydła na 4 części dwie (prawa-lewa) zewnętrzne, z lotkami i dwie wewnętrzne z hamulcami aerodynamicznymi i klapami (zależnie od symulowanego modelu szybowca). Do tego kadłub, ster kierunku i ster wysokości.

- masa
- powierzchnia
- położenie środka elementu względem środka masy
- stan mechanizacji skrzydła (nie dotyczy kadłuba)
- informacje dotyczące momentu bezwładności elementu

Informacje te są niezbędne do wyliczenia wszelkich danych symulacji – od masy całkowitej, przez wielkość siły nośnej.



Rys. 4.2 Podział szybowca

---

## 2. POBRANIE DANYCH DOTYCZĄCYCH WYDAJNOŚCI AERODYNAMICZNEJ.

W tym punkcie odwołujemy się do danych opisanych w rozdziale 2.2.2, gdzie dzięki programowi XFOIL otrzymaliśmy tabele współczynników  $C_D(\alpha)$  i  $C_L(\alpha)$  dla konkretnych profili lotniczych wykorzystanych w symulacji. Uwzględniamy zastosowanie różnych profili w różnych modelach samolotów.

## 3. ZSUMOWANIE SIŁ, DODANIE DZIAŁANIA GRAWITACJI I WYLICZENIE MOMENTÓW

Tutaj na podstawie dostarczonych danych policzone zostają wszelkie siły i momenty działające na poszczególne elementy samolotu. Dodawane jest oddziaływanie ruchów powietrza. Następnie wyniki sumowane są dla całego szybowca. Tutaj też dokłada się oddziaływanie grawitacji.

## 4. OBLICZENIE PRZYSPIESZEŃ I PRĘDKOŚCI

Ten fragment programu odpowiada za wyliczenie z odpowiednich wzorów i danych dostarczonych przez punkt 1 oraz 3. przyspieszenia liniowego i kątownego, dzięki czemu w następnym punkcie korzystając z metody Eulera i poprzez przekształcenie kwaternionowe wyliczyć można przemieszczenia i rotacje szybowca.

## 5. PRZEMIESZCZENIE I OBRÓT MODELU

Na końcu odbywa się przekazanie wyliczonych wartości obrotu i przemieszczenia do wyjścia skryptu oraz przekazanie ich do zegarów szybowca. Z tego też miejsca biorą się dane wejściowe do wcześniejszych punktów, a więc kąt natarcia z rotacji do punktu 2., niezbędny do pobrania odpowiednich współczynników  $C_D$  i  $C_L$ , a także prędkości, aby móc później wyliczyć wartość sił oporu i siły nośnej.

Punkty od 2 do 6 są wykonywane w każdym kroku symulacji, co równoznaczne jest z klatką symulacji, a więc dzieje się 24 razy na sekundę.

## 4.2 Środowisko programistyczne

Model fizyczny skonstruowany został jako skrypt obsługiwany przez środowisko Softimage. Posiada ono wsparcie dla trzech języków skryptowych:

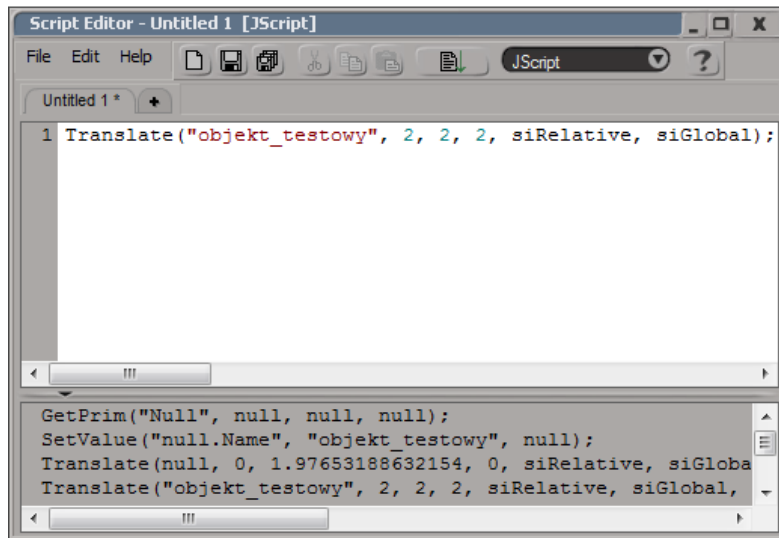
- **JScript**
  - Opracowany przez firmę Microsoft w oparciu o język JavaScript
  - Model obiektowy
  - Brak różnic między wskaźnikami i zmiennymi
  - Większe uporządkowanie i prostota jak VBScript
  - Podobny do C++, co powoduje łatwiejszą konwersję
  - Dobra obsługa błędów
  - Dobra dokumentacja internetowa
- **VBScript**
- **Python**

Rodzaje programowania w środowisku Softimage:

- **Skrypty**
  - Statyczne
  - Wykonywane jednokrotnie
- **Wyrażenia**
  - Niestatyczne
  - Nie mogą wykorzystywać niektórych funkcjonalności skryptów
  - Służą do wykonywania prostych operacji na obiektach
- **Operatory skryptowe**
  - Niestatyczne
  - Mogą wykorzystywać większość funkcjonalności skryptów

*Softimage* posiada panel dedykowany do pisania skryptów – edytor skryptów (Rys. 4.3). Edytor posiada dwa obszary: okno historii oraz edycji skryptów. To pierwsze, występujące na samym dole edytora, wyświetla historię ostatnio wywoływanych poleceń. Co ważne polecenia te nie są tylko komendami wywoływanymi z edytora skryptów. Każda czynność wykonana w środowisku 3D także prowadzi do pojawienia się odpowiedniej komendy w oknie historii. Przykładowo przesunięcie aktywnego obiektu na scenie poprzez użycie narzędzia „*translate tool*” spowoduje pojawienie się polecenia „*Translate(...)*”. Okno edytora skryptów służy natomiast do pisania własnych poleceń, które następnie mogą zostać wywołane. Edytor skryptów posiada także pasek narzędzi (powyżej wymienionych wcześniej obszarów), z poziomu którego możemy wykonywać czynności takie jak: zapis, odczyt, uruchomienie lub otwarcie nowego pliku skryptu. Pozwala także ustawić język programowania na jeden z wcześniej wymienionych. Co ważne zmiana języka powoduje

automatyczne przekształcanie wszystkich komend występujących w oknie historii na wybrany język.

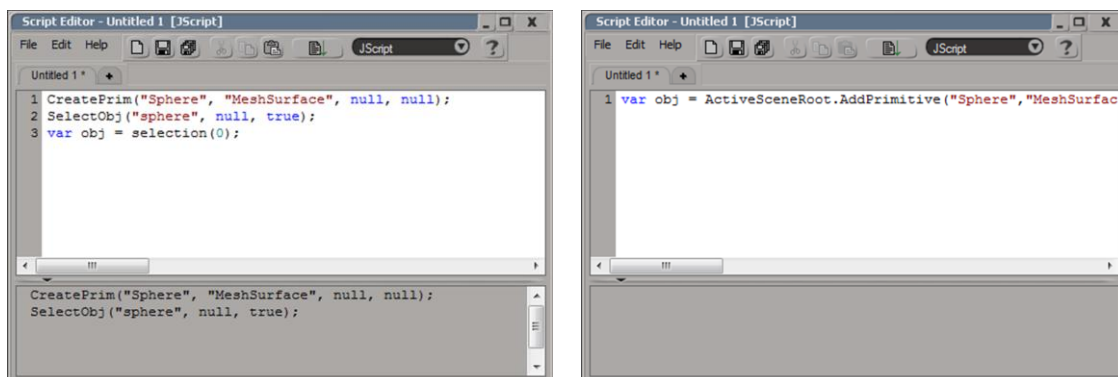


Rys. 4.3 Okno edytora skryptów

Pisząc skrypty w środowisku Softimage niezależnie od języka programowania mamy do dyspozycji dwa modele: komend oraz obiektowy (patrz **Błąd! Nie można odnaleźć źródła odwołania.**). Model komend charakteryzuje się wykorzystaniem funkcji wbudowanych do zarządzania elementami sceny. Modelem tym posługuje się XSI w momencie wykonywania czynności poza edytorem skryptów, co powoduje, że czynności te pojawiają się następnie w oknie historii w postaci poleceń. Model ten wykorzystywany jest do tworzenia prototypów oraz mniejszych skryptów. Model obiektowy polega na wykorzystaniu obiektów, ich metod i właściwości, jest bardziej funkcjonalny i pozwala osiągać te same rezultaty przy użyciu

### Przykład zastosowania obu modeli programistycznych

Za przykład posłużmy nam proces tworzenia obiektu sfery siatki wielokątów (*polygon mesh* → *Sphere*) oraz przypisanie jej do zmiennej.



a) model komend

b) model obiektowy

Rys. 4.4 Przykład zastosowania obu modeli programistycznych

## Przykład zastosowania obu modeli programistycznych c.d.

### Model komend

```
CreatePrim("Sphere", "MeshSurface", null, null);
SelectObj("sphere", null, true);
var obj = selection(0);
```

- `CreatePrim(PresetObj, GeometryType, [Name], [Parent])`; tworzy na scenie odpowiedni prymityw, a następnie konwertuje go do określonego typu geometrii. Przyjmuje następujące parametry:
  - `PresetObj` – string opisujący prymityw odpowiedniego typu, w naszym przypadku jest to sfera
  - `GeometryType` – string opisujący odpowiednią geometrie, u nas jest to powierzchnia typu siatki wielokątów („*Polygon Mesh Surface*”)
  - `Name` – nazwa obiektu, w przykładzie przekazaliśmy w tym parametrze `null`, co powoduje przypisanie domyślnej nazwy „*sphere*”
  - `Parent` – określenie rodzica tworzonego obiektu, `null` powoduje ustawienie jako rodzica obiektu korzenia sceny („*Scene Root*”)
- `SelectObj([SelectionList], [HierarchyLevel], [CheckObjectSelectability])`; - dokonuje uaktywnienia elementów sceny. Posiada parametry:
  - `SelectionList` – lista nazw elementów do zaznaczenia
  - `HierarchyLevel` – określa sposób zaznaczania obiektów w hierarchii
  - `CheckObjectSelectability` – określa czy zaznaczać obiekty niezaznaczalne
- `selection(0)`; - zwraca pierwszy z aktywnych obiektów

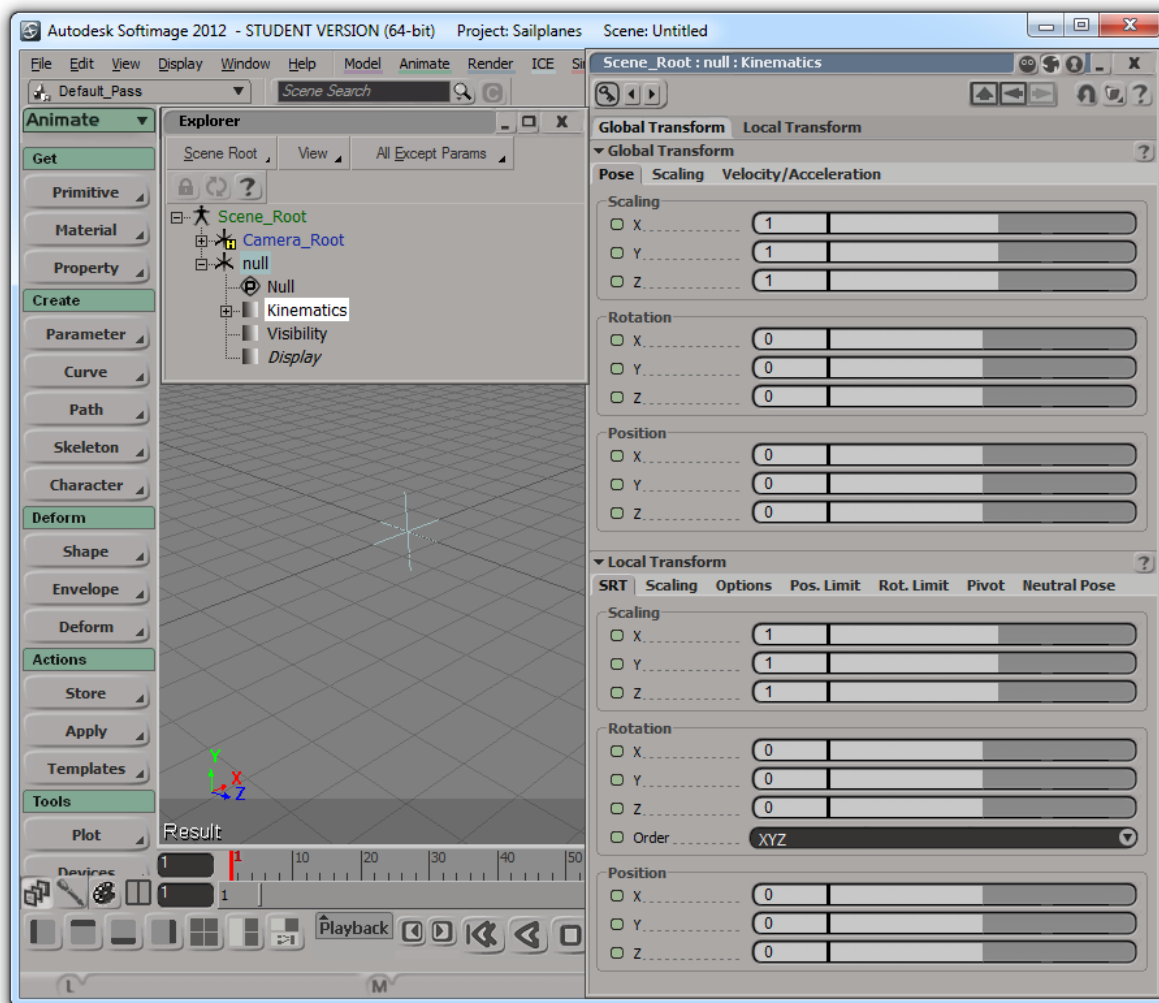
### Model obiektowy

```
var obj = ActiveSceneRoot.AddGeometry("Sphere", "MeshSurface");
```

- `X3DObject.AddGeometry(Preset, [Type], [Name])`; metoda `AddGeometry` obiektu `X3DObject`, która tworzy nowy prymityw na scenie, a także zwraca ten element, dzięki czemu możemy w jednej linii dokonać przypisania nowo utworzonego obiektu do zmiennej. Metoda przyjmuje parametry:
  - `Preset` – nazwa presetu opisująca rodzaj prymitywu jaki chcemy utworzyć
  - `Type` – typ geometrii
  - `Name` – nazwa potomka hierarchii

krótszych, wydajniejszych algorytmów. Wykorzystywany jest przy tworzeniu operatorów skryptowych.





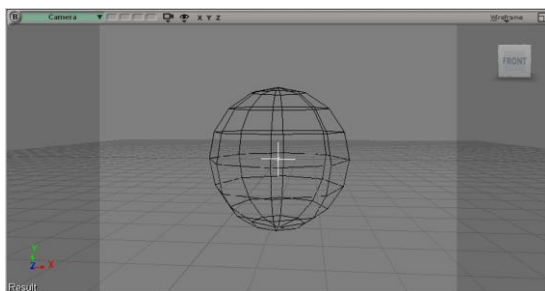
Rys. 4.5 Właściwości kinematyczne obiektu typu *null*

Każdy element sceny 3D środowiska *Softimage* posiada swoje właściwości kinematyczne (Rys. 4.5) które pozwalają zarządzać między innymi położeniem, rozmiarem i kształtem elementu, przy czym ilość właściwości i ich złożoność różni się w zależności od rodzaju elementu. Właściwości te posiadają możliwości zmiany wartości w czasie na przykład poprzez wstawianie klątek kluczowych w określonych chwilach i odpowiednią interpolację wartości pośrednich. Funkcjonalność tą wskazuje zielony, żółty lub czerwony kwadrat przy nazwie właściwości przy czym kolor zależy od ewentualnego występowania klątki kluczowej danego parametru w aktualnej chwili czasowej. Inną możliwością zarządzania wartościami parametrów zmiennych w czasie są wyrażenia. Dzięki wyrażeniom możemy zarządzać wartościami dynamicznie przy użyciu skryptu zamiast poprzez ustawianie zafiksowanych w czasie wartości. Dla każdego parametru kontrolowalnego możemy utworzyć wyrażenie skryptowe kontrolujące jego zmiany (patrz **Błąd! Nie można odnaleźć źródła odwołania.**).

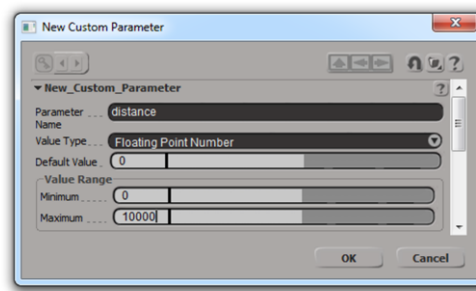
## Przykład zastosowania wyrażeń

Dobrym przykładem zastosowania wyrażeń może być algorytm, który powoduje uzależnienie złożoności elementu sceny od odległości kamery. Spowoduje to, że obiekt ten będzie wyglądał lepiej w trakcie zbliżeń, czyli wtedy gdy jest to najbardziej potrzebne, natomiast w trakcie oddalania kamery jego rozdzielczość będzie malała co spowoduje oszczędność mocy obliczeniowej.

W pierwszej kolejności tworzymy na scenie obiekt, którego złożoność będzie sterowana odległością kamery, w naszym przypadku jest to sfera. Tworzymy także obiekt *null* (Rys. 4.6a), a następnie wiążemy jego położenie do położenia kamery.



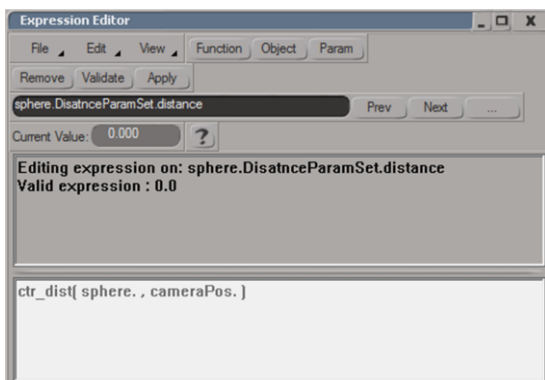
a) utworzone obiekty



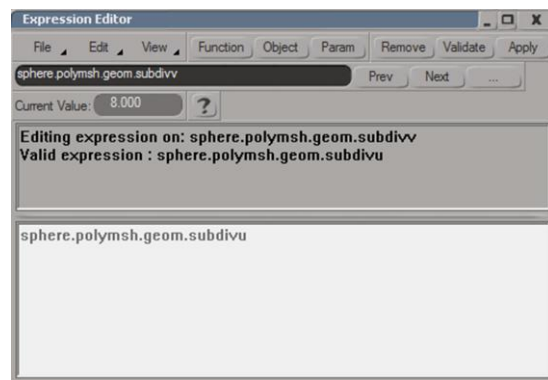
b) dodawanie własnego parametru

Rys. 4.6 Przygotowanie sceny

Następnie skorzystamy z jednego z bardzo potężnych konceptów środowiska Softimage jakim jest możliwość tworzenia własnych zestawów parametrów. Funkcjonalność ta pozwala na tworzenie własnych okien GUI gromadzących parametry istniejące już w środowisku jak np. wybrane właściwości obiektów sceny. Wtedy utworzony przez nas zestaw parametrów pełni funkcję grupującą. Istnieje także możliwość tworzenia zupełnie nowych parametrów (zmiennych) o określonym typie. Do wyboru mamy zmienną tekstową, logiczną, zmiennoprzecinkową i całkowitą. W naszym przypadku tworzymy zestaw parametrów posiadający tylko jeden parametr typu zmiennoprzecinkowego – dystans (Rys. 4.6b), który będzie przechowywał informacje jak odległe są środki utworzonych wcześniej obiektów. Ustawiamy zakres zmiennej od 0 do 10000, określając w ten sposób przedział wewnątrz, którego na pewno znajdują się wszystkie stosowane przez nas położenia kamery. Następnie przypisujemy wyrażenie do utworzonego parametru (Rys. 4.7a). Zastosowanie tego wyrażenia powoduje, że do wartości *distance* przypisywana jest odległość między sferą i kamerą (Rys. 4.8).



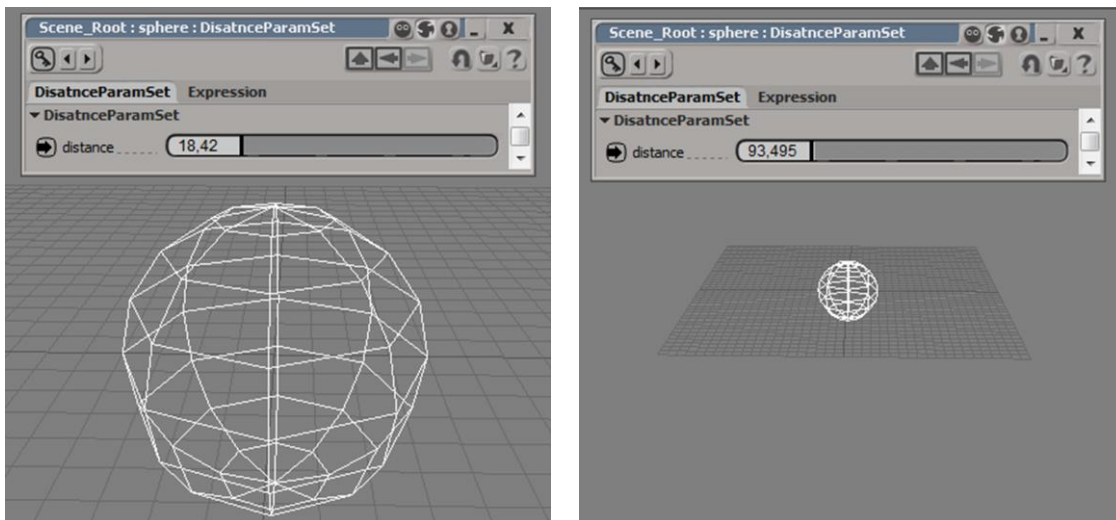
a) Wyrażenie dla parametru *distance*



b) Wyrażenie dla parametru *v* geometrii sfery

Rys. 4.7 Stosowanie edytora wyrażeń

### Przykład zastosowania wyrażeń c.d.

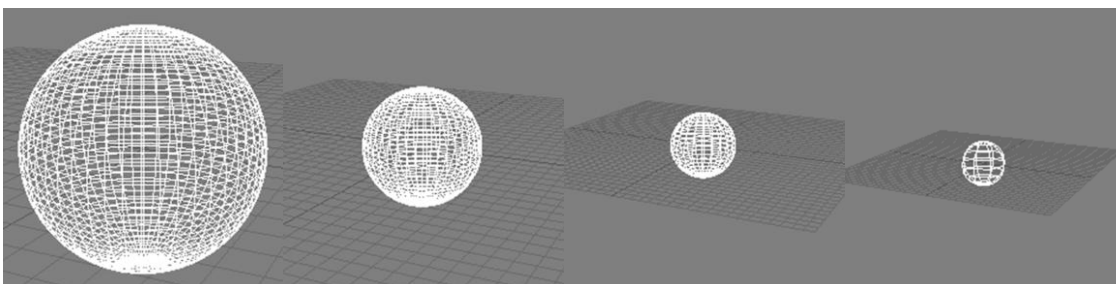


a) odległość 18,42

b) odległość 93,495

Rys. 4.8 Zmiana parametru *distance* w zależności od odległości kamery od obiektu

Przechodzimy do parametrów podziałów geometrii sfery w kierunkach lokalnych *uv*. Jeden z nich (wybraliśmy *u*) linkujemy do parametru *distance*. Linkowanie takie powoduje przypisanie domyślnej wartości podziałów (8 podziałów) do aktualnej wartości parametru *distance* czyli odległości kamery od obiektu. Należy więc pamiętać, że w trakcie linkowania kamera musi być ustawiona w satysfakcjonujący nas sposób (ponieważ linkujemy dla minimalnej przyjmowanej rozdzielczości musi znajdować się daleko od obiektu). Dla kierunku *v* chcemy otrzymać taki sam efekt. W tym celu wystarczy zastosować wyrażenie przepisujące wartość *u* do wartości parametru *v* (Rys. 4.7b). Następnie przybliżamy kamerę bardzo blisko obiektu, odpowiednio zwiększamy ilość podziałów i ustawiamy *Set Relative Values* w opcjach linkowania. W ten sposób dla odpowiedniej odległości ustawione zostały wartości maksymalne i minimalne ilości geometrii obiektu. Wszystkie wartości pośrednio są interpolowane, a zmiana odległości kamery bezpośrednio wpływa na złożoność geometrii obiektu (Rys. 4.9).

a) dystans 21  
złożoność 50b) dystans 43  
złożoność 39c) dystans 79  
złożoność 22d) dystans 114  
złożoność 8

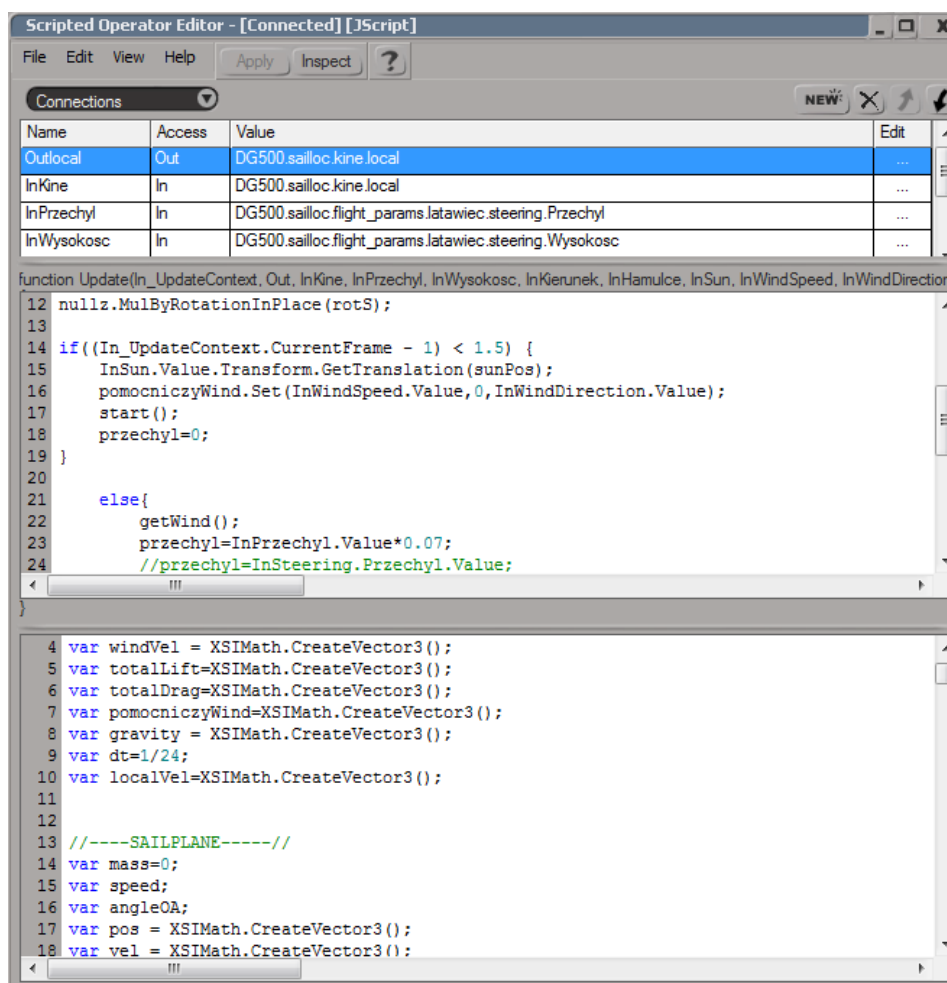
Rys. 4.9 Efekt końcowy

## OPERATORY SKRYPTOWE

Większość utworzonej przez nas fizyki lotu obsługiwana będzie przy pomocy trzeciego sposobu tworzenia skryptów w środowisku Softimage, którym są operatory skryptowe. Podyktowane jest to tym, że podobnie jak wyrażenia operatory skryptowe są niestaticzne. Możemy bezpośrednio wpływać na efekt działania skryptu wywoływanego operatorem w trakcie jego działania. Stwarza to więc możliwość kontroli szybowca w czasie rzeczywistym. Do tego w odniesieniu do wyrażen operatory skryptowe posiadają możliwość tworzenia bardziej rozbudowanego kodu dodając wiele funkcjonalności:

- tworzenia metod i pętli programistycznych
- deklarowanie zmiennych lokalnych
- robienie wcięć w kodzie i dodawania komentarzy

Następną zaletą operatorów skryptowych jest sposób ich implementacji. Na rysunku (Rys. 4.10) przedstawione zostało okno edytora operatorów skryptowych. Podzielone jest ono na trzy obszary.



The screenshot shows the 'Scripted Operator Editor' window. At the top, there is a menu bar (File, Edit, View, Help) and buttons for 'Apply', 'Inspect', and a help icon. Below the menu is a 'Connections' table:

Name	Access	Value	Edit
Outlocal	Out	DG500.sailloc.kine.local	...
InKine	In	DG500.sailloc.kine.local	...
InPrzechyl	In	DG500.sailloc.flight_params.latawiec.steering.Przechyl	...
InWysokosc	In	DG500.sailloc.flight_params.latawiec.steering.Wysokosc	...

Below the table, there are two sections of code. The first section is a function definition:

```
function Update(In_UpdateContext, Out, InKine, InPrzechyl, InWysokosc, InKerunek, InHamulce, InSun, InWindSpeed, InWindDirection)
12 nullz.MulByRotationInPlace(rotS);
13
14 if(In_UpdateContext.CurrentFrame - 1) < 1.5) {
15     InSun.Value.Transform.GetTranslation(sunPos);
16     pomocniczyWind.Set(InWindSpeed.Value, 0, InWindDirection.Value);
17     start();
18     przechyl=0;
19 }
20
21 else{
22     getWind();
23     przechyl=InPrzechyl.Value*0.07;
24     //przechyl=InSteering.Przechyl.Value;
    ...
}
```

The second section is a list of variable declarations:

```
4 var windVel = XSIMath.CreateVector3();
5 var totalLift=XSIMath.CreateVector3();
6 var totalDrag=XSIMath.CreateVector3();
7 var pomocniczyWind=XSIMath.CreateVector3();
8 var gravity = XSIMath.CreateVector3();
9 var dt=1/24;
10 var localVel=XSIMath.CreateVector3();
11
12
13 //-----SAILPLANE-----//
14 var mass=0;
15 var speed;
16 var angle0A;
17 var pos = XSIMath.CreateVector3();
18 var vel = XSIMath.CreateVector3();
```

Rys. 4.10 Okno edytora operatora skryptowego

Pierwszym z nich jest znajdujące się na górze pole *Connections/Variables*, gdzie definiować możemy porty wejścia/wyjścia oraz zmienne lokalne naszego operatora skryptowego. Porty reprezentują miejsca, w których operator komunikuje się ze środowiskiem *Softimage*. W ich miejsce podłączać można różne rodzaje danych jak chociażby obiekty, ich właściwości lub parametry. W naszym przypadku będą to dane o położeniu układów sterowniczych szybowca.

Porty wejścia /wyjścia oraz zmienne ustawione w pierwszym oknie przekazywane są do funkcji *Update*, której treść znajduje się w środkowym obszarze edytora. Tutaj zamieszczać możemy kod funkcji, która wywoływana będzie przy każdej zmianie, któregoś z parametrów wejściowych lub w razie potrzeby zwrócenia któregoś z parametrów wyjściowych. Dodatkowym zestawem danych przekazywanym do funkcji jest tzw. *In\_UpdateContext* określający kontekst w jakim znajduje się operator. Zestaw ten zawiera wiele przydatnych informacji jak chociażby aktualna klatka animacji.

Najniższy obszar edytora pozwala zdefiniować instrukcje dodatkowe, wykonywane jednorazowo, przy parsowaniu skryptu. Jest to więc miejsce gdzie definiujemy zmienne i funkcje pomocnicze, na których operować będzie funkcja *Update*.

## 4.3 Opis programu



### 4.3.1 Opis panelu sterowania, wejścia i wyjścia skryptu.

Na wejściu do skryptu, w sekcji połączeń (*Connections*) edytora skryptowego podajemy następujące wartości (Rys. 4.11):

- Wyjście /Out/ - OutLocal - położenie szybowca w lokalnym układzie współrzędnych
- Wejście /In/ - InKine - położenie szybowca w lokalnym układzie współrzędnych

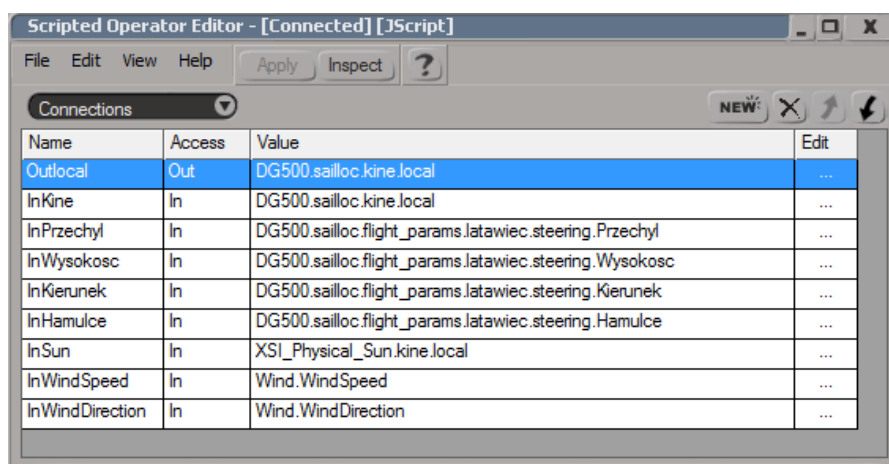
Kolejne wejścia służące do sterowania szybowcem to:

- InPrzechyl – sterowanie lotkami
- InWysokosc – sterowanie sterem wysokości
- InKierunek – sterowanie sterem kierunku
- InHamulce – sterowanie hamulcami aerodynamicznymi

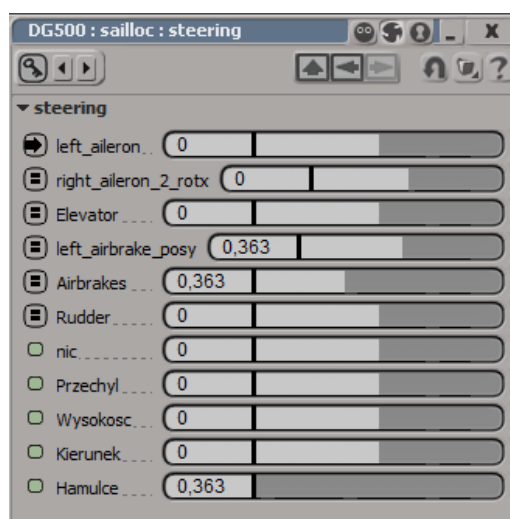
Wejścia służące do sterowania warunkami atmosferycznymi:

- InSun – wejście dające informacje o pozycji słońca
- InWindSpeed, InWindDirection – wejście dotyczące sterowania wiatrem

Połączenie symulacji graficznej – a więc obrotu czy przesunięcia elementów mechanizacji skrzydeł z modyfikacją danych wejściowych dzieje się przez stworzony *Custom Parameter Set* pokazany na Rys. 4.12. Widać tam, idąc od góry sześć parametrów sterujących elementami mechaniki skrzydeł, następnie jeden parametr testowy „*nic*”, a na końcu cztery parametry powiązane do połączeń (*Connections*) skryptu. Ich nazwy odpowiadają nazwom wejść do skryptu. Powodem takiego skonstruowania panelu sterowania jest konieczność rozdzielenia sterowania skrytem i kinematyką elementów budowy szybowca. Parametry podwiązane do skryptu połączone są z parametrami sterującymi elementami mechaniki przy użyciu wyrażeń skryptowych. Stąd też strzałka przy parametrze `left_aileron` oznacza, że wartość tego parametru (obrotu lewej lotki) jest wyliczana za pomocą przypisanego równania (zapisane tam równanie to `left_aileron=(-1)*right_aileron`). Ponieważ pochylenie drążka sterowania w szybowcu w bok powoduje jednocześnie wychylenie jednej lotki w górę a drugiej w dół, tak samo u nas są one powiązane i kąt wychylenia jednej jest przeciwnością kąta wychylenia drugiej. Pozostałe parametry sterowania elementami szybowca posiadają znaki równości, co oznacza że są tam przypisane proste zależności bez dodatkowych obliczeń. I tak zmiana danych wejściowych do skryptu powoduje jednocześnie zmianę w ustawieniu elementów konstrukcyjnych szybowca.



Rys. 4.11 Połączenia operatora skryptowego



Rys. 4.12 Panel sterowania szybowcem

### 4.3.2 Obliczenia przed startem symulacji.

Najkorzystniejszym rozwiązaniem pod kątem wydajności działania programu jest wykonanie jak największej ilości obliczeń jeszcze przed rozpoczęciem symulacji, aby nie wykonywać niepotrzebnych obliczeń w trakcie jej działania. Dlatego też konstruując nasz operator skryptowy staraliśmy się jak najwięcej obliczeń umieścić w dolnym panelu edytora operatorów skryptowych, który to wywoływany jest przy jego inicjalizacji (patrz **Błąd! Nie można odnaleźć źródła odwołania.**). W tym właśnie miejscu dokonujemy podziału szybowca na 7 elementów, które zapisane są w tablicy obiektów zawierających wszelkie potrzebne parametry do obliczenia równań ruchu.

Dwie poniższe ilustracje pokazują: prototyp funkcji inicjalizującej elementy szybowca (Rys. 4.13) i obliczenie tensora momentu bezwładności oraz jego odwrotności (Rys. 4.14).

```
var Elements = new Array(7);
function element (area,dx,dy,dz,weight,steering,mInertiaX,mInertiaY,mInertiaZ) {
    this.area=area;
    this.dx=dx;
    this.dy=dy;
    this.dz=dz;
    this.weight=weight;
    this.steering=steering;
    this.mInertiaX=mInertiaX;
    this.mInertiaY=mInertiaY;
    this.mInertiaZ=mInertiaZ;
}
```

Rys. 4.13 Inicjalizacja tablicy elementów konstrukcyjnych szybowca

```
for(var i=0;i<7;i++) mass+=Elements[i].weight;
var Ixx = 0;    var Iyy = 0;    var Izz = 0;
var Ixy = 0;    var Ixz = 0;    var Iyz = 0;
for (var i = 0; i < 7; i++)
{
    Ixx += Elements[i].mInertiaX + Elements[i].fMass * (Elements[i].dy*Elements[i].dy
    + Elements[i].dz*Elements[i].dz);
    Iyy += Elements[i].mInertiaY+ Elements[i].fMass * (Elements[i].dz*Elements[i].dz
    + Elements[i].dx*Elements[i].dx);
    Izz += Elements[i].mInertiaZ + Elements[i].fMass * (Elements[i].dx*Elements[i].dx
    + Elements[i].dy*Elements[i].dy);
    Ixy += Elements[i].weight * (Elements[i].dx * Elements[i].dy);
    Ixz += Elements[i].weight * (Elements[i].dx * Elements[i].dz);
    Iyz += Elements[i].weight * (Elements[i].dy * Elements[i].dz);
}
mOInertia.Set( Ixx,-Ixy,-Ixz,
               -Ixy,Iyy,-Iyz,
               -Ixz,-Iyz,Izz);
mOInertiaI.Invert(mOInertia);
```

Rys. 4.14 Obliczenie elementów macierzy tensora momentu bezwładności

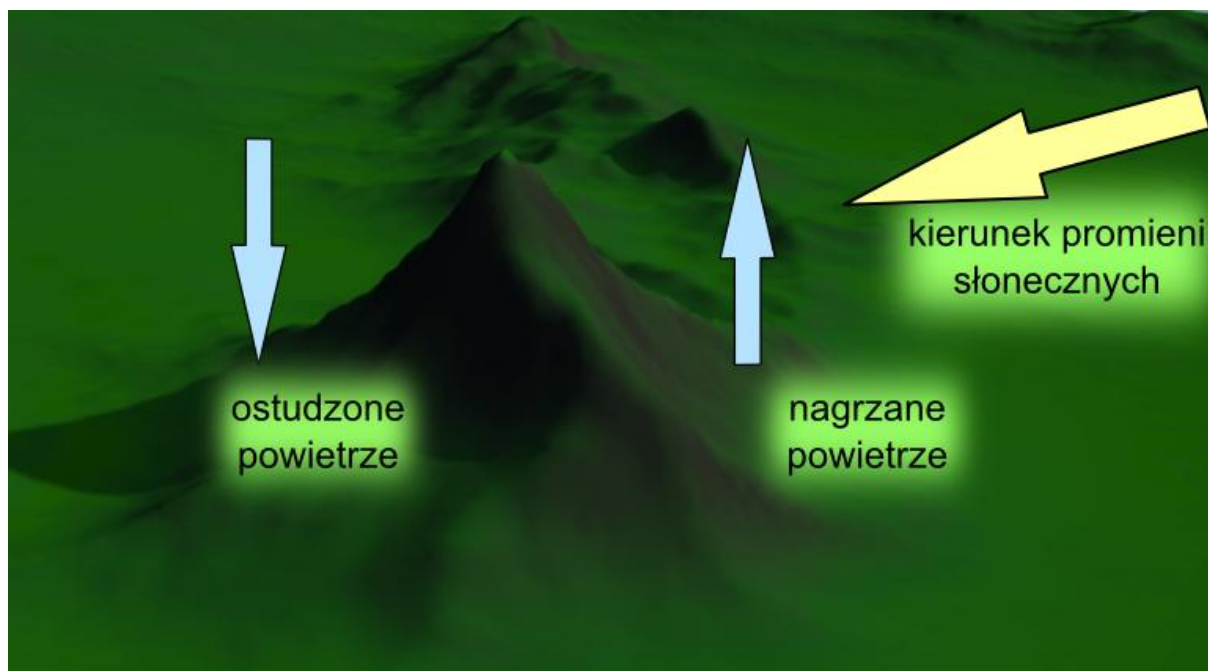
Dodatkowo tutaj implementowane są tablice współczynników  $C_D$  i  $C_L$  pobrane z programu *XFOIL* a także obliczony zostaje kwaternion obrotu. Wyliczona zostaje także tablica przechowująca wektory prędkości wiatru.

### 4.3.3 Warunki atmosferyczne

Warunki atmosferyczne działające na szybowiec to wiatr i prądy termiczne. Ich wpływ na lot szybowca jest stosunkowo prosto wprowadzić – wystarczy podać „wektor wiatru”, zawierający składowe poziome i pionową ruchów powietrza. Następnie w miejscu symulacji, w którym obliczamy wartości sił aerodynamicznych, do wektora prędkości szybowca dodajemy wektor prędkości wiatru. Bardziej złożonym zadaniem jest jednak wyznaczenie



wartości tego wektora. Zdecydowaliśmy się przyjąć model, w którym ruchy termiczne zależą od nasłonecznienia podłoża. W skrócie – nasłoneczniony fragment zbocza, będzie się szybko nagrzewać ogrzewając powietrze znajdujące się nad nim i powodując wystąpienie prądów termicznych idących w górę. Odwrotnie będzie to wyglądać dla zacienionej części zbocza (Rys. 4.15). Ponieważ w symulacji lot odbywać się będzie po terenie górzystym, takie rozwiązanie da nam zróżnicowane warunki atmosferyczne dla całej mapy.



Rys. 4.15 Wpływ słońca na występujące prądy powietrzne

W celu wyznaczenia rozkładu temperatury podłoża, za pomocą którego umiejscawiamy wznoszące i opadające prądy termiczne, wyliczamy kąt, jaki wektor normalny podłoża tworzy z wektorem skierowanym od słońca w stronę podłoża. Następnie obliczana jest wartość iloczynu skalarnego znormalizowanego wektora normalnego i znormalizowanego wektora od słońca do powierzchni. Uzyskaną wartość wystarczy dowolnie przeskalować, aby uzyskać prędkość prądów w granicach kilku metrów na sekundę. Wyniki zapisywane są w tabeli, której każdy element odpowiada konkretnemu miejscu na rzucie mapy na płaszczyznę. Kiedy szybowiec znajduje się najbliżej danego węzła – pobierane są dane z tabeli odpowiadające temu miejscu i tak wyznacza się wpływ prądu termicznego na ruch szybowca. Obliczenia danych do tabeli dokonywane są w trakcie inicjalizacji, aby zmniejszyć ilość obliczeń wykonywanych w każdym kroku. Sterowanie poziomą składową prędkości wiatru odbywa się za pomocą specjalnie utworzonego *Custom Parameter Set*'u nazwanego *Wind* (Rys. 4.16). Daje nam on możliwość wybrania na początku prędkości wiatru z zakresu (0;20)m/s a także kierunku wiatru przez wybranie kąta z zakresu (0, 360) stopni.



Rys. 4.16 Sterowanie wiatrem

### 4.3.4 Opis warunków początkowych

Warunkami początkowymi symulacji będzie ustawienie szybowca w odpowiednim punkcie w przestrzeni – wybieramy wysokość i miejsce startu, ustawienie jego orientacji (rotacji) i szybkości początkowej (wektor prędkości wyliczony zostanie na podstawie szybkości i rotacji). Dodatkowo mamy ustalone nasłonecznienie terenu zmienne w zależności od nachylenia powodujące pionowe ruchy powietrza (prądy termiczne) a także poziome ruchy powietrza zależne od ustawienia wiatru.

### 4.3.5 Krok symulacji

W każdym z kroków symulacji wywoływane będą funkcje obliczające działające siły, a następnie wyliczające przyspieszenia liniowe i kątowe, aby w końcu przekazać odpowiednie wartości przesunięć i obrotów na wyjście, jak i do następnej iteracji programu, gdzie takie dane jak prędkość czy kąt natarcia są niezbędne do wyliczenia siły nośnej. Tutaj też dokładniej opiszemy schemat obliczania powstałych sił aerodynamicznych. Dla przykładu skupimy się na skrzydłach:

- Pobranie wektora przesunięcia szybowca i wektora zwrotu dzioba szybowca
- Dla każdego elementu osobno, do wektora prędkości dodawany jest wektor wiatru pobrany dla danego miejsca na mapie, a także czynnik zależny od obrotu szybowca ( $\boldsymbol{\omega} * \boldsymbol{r}$ ) – iloczyn prędkości kątowej i odległości od środka obrotu (gdy szybowiec skręca w prawo, lewe skrzydło porusza się względem powietrza szybciej niż prawe, co powoduje wzrost siły nośnej po lewej stronie i przechył szybowca na prawą stronę)
- Kąt między powyższymi wektorami to kąt natarcia – dla niego wyszukiwany jest w tablicy utworzonej z danych z programu XFOIL odpowiedni współczynnik  $C_D$  i  $C_L$
- Obliczana jest wartość siły nośnej  $L$  i oporu  $D$  dla każdego z elementów po uprzednim sprawdzeniu stanu mechanizacji skrzydła, który jeśli nie jest w neutralnej pozycji dodatkowo modyfikuje współczynnik  $C_D$  i  $C_L$

Po wyliczeniu sił obliczane są odpowiednie momenty, uaktualniony zostaje kwaternion obrotu i dane przekazywane są na wyjście skryptu powodując przemieszczenie i rotację modelu szybowca.

```

function krokCalosci() {
    forceVector.SetNull();
    momentsVector.SetNull();
    angleOA=vel.Angle(nullz);
    rot2L.SetFromXYZAnglesValues(-1.57,rotS.XYZAngles.y,rotS.XYZAngles.z);
    for(var i=0;i<6;i++)
    {
        krokElementu(Elements[i].area,Elements[i].dx,Elements[i].dy,Elements[i].dz);
        forceVector.AddInPlace(elementForce);
        momentsVector.AddInPlace(elementMoments);
    }
    forceVector.AddInPlace(gravForce);
    acceleration.Scale(1/mass,forceVector);
    dv.Scale(dt,acceleration);
    vel.AddInPlace(dv);
    movement.Scale(dt,vel);
    pos.AddInPlace(movement);
}

```

Rys. 4.17 Funkcja „krokCalosci”, wywołująca funkcję „krokElementu” dla każdego z elementów szybowca i nadająca wartość wektorowi przesunięcia

```

function krokElementu(pole,x,y,z)
{
    elementForce.SetNull();
    elementMoments.SetNull();
    R.Set(x,y,z);
    localVel=vel;
    localVel.AddInPlace(windVel);
    localVel.ScaleAdd(R.Length(),angularVelo,localVel);
    LVector.MulByRotation(vel,rot2L);
    LVector.NormalizeInPlace();
    DVector.Copy(vel);
    DVector.NormalizeInPlace();
    DVector.ScaleInPlace(-1*countD(angleOA,pole,localVel.Length()));
    LVector.ScaleInPlace(countL(angleOA,pole,localVel.Length()));
    elementForce.AddInPlace(LVector);
    elementForce.AddInPlace(DVector);
    elementMoments.Cross(elementForce,R);
}

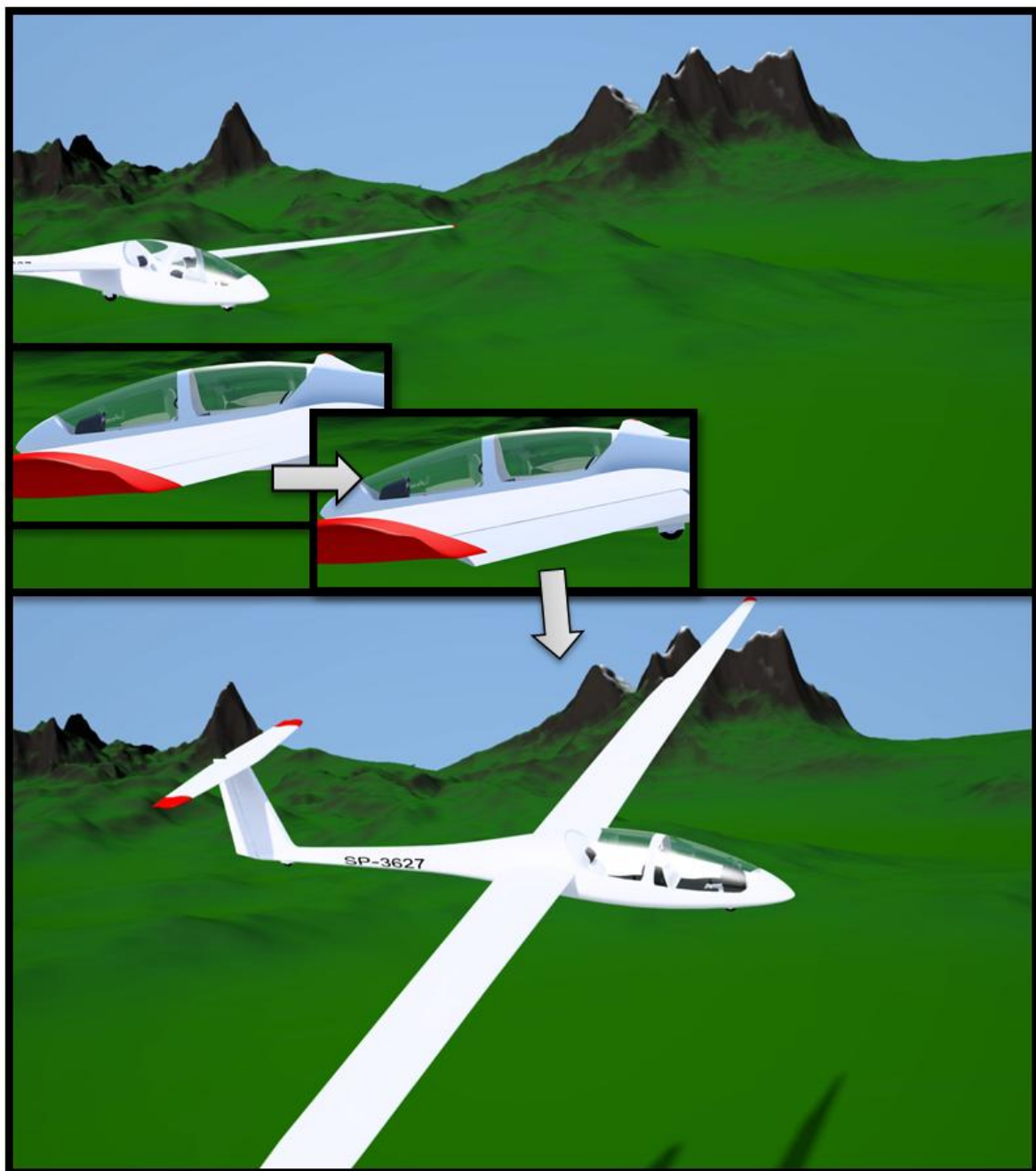
```

Rys. 4.18 Funkcja „krokElementu”, gdzie wyliczany jest wektor prędkości dla konkretnego elementu i obliczane są składowe siły działające na niego

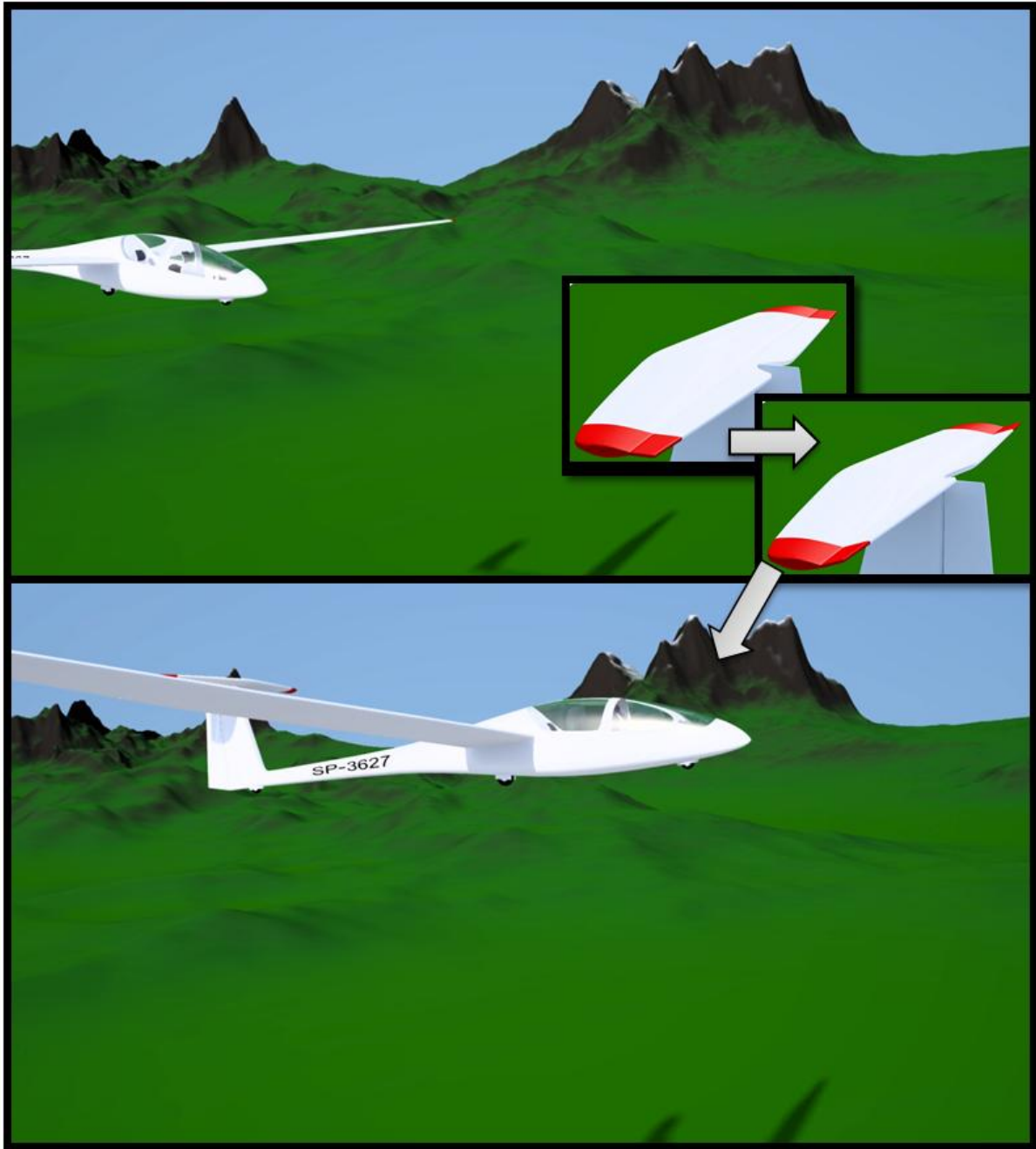
## 4.4 Przykładowe wyniki działania

Wynikiem działania skryptu jest symulacja przedstawiająca szybowiec zdolny do utrzymania się w powietrzu dzięki wyliczonym w programie siłom aerodynamicznym równoważącym ustaloną siłę grawitacji. Potrafi on skręcać i obracać się, wznosić i opadać przez zmianę zależnych od kąta natarcia sił. Dodatkowo oddziałuje on z otoczeniem – wiatrem i słońcem wywołującym ruchy termiczne.

Przechyl szybowca odbywa się przez wychylenie lotek, które zwiększają po jednej, a zmniejszają po drugiej stronie siłę nośną skrzydła – powodując powstanie momentu siły na osi biegnącej w poprzek szybowca.



Wznoszenie powodowane jest zwiększaniem siły nośnej poprzez wzrost kąta natarcia. Wzrost kąta natarcia spowodowany jest wychyleniem steru wysokości zmniejszającym wytwarzaną siłę nośną, lub też kierując ją w dół. Tak powstaje moment sił na osi biegnącej od przodu do tyłu modelu.



Należy mieć na uwadze, że uzyskanych przez nas wyników nie da się w pełnej okazałości przedstawić w formie papierowej. Pełna wartość naszej pracy uwidacznia się dopiero po obejrzeniu szybowca znajdującego się w ruchu.

## 5 Podsumowanie

W nowoczesnej grafice komputerowej nie wystarcza już skupienie się na estetyce i wiernym odwzorowaniu wyglądu modelowanych obiektów. W tym momencie najistotniejsze dla grafików wydaje się być odwzorowanie zachowania. Aby to zrobić, niezbędne jest dogłębne poznanie praw fizyki rządzących danym zjawiskiem oraz poznanie struktury symulowanego obiektu. Stąd też przy nowoczesnych projektach z dziedziny grafiki komputerowej cała praca przebiega dwutorowo – z jednej strony graficy skupiają się na odwzorowaniu wyglądu obiektu, a z drugiej fizycy i programiści próbują zasymulować mechanizmy rządzące ruchem i zachowaniem obiektu.

Można powiedzieć, że nasz dwuosobowy zespół doskonale odzwierciedla kierunek rozwoju i sposób działania w nowoczesnej grafice komputerowej. Po zdecydowaniu się na powyższy temat niejako podzieliliśmy się obowiązkami i jeden z nas skupił się na zdobywaniu wiedzy i umiejętności z grafiki komputerowej pojętej jako modelowanie wyglądu, a drugi z nas skupił się na zagadnieniach aerodynamiki i inżynierii lotniczej. Oczywiście w trakcie pisania pracy i tworzenia modelu i symulacji podzieliliśmy się wiedzą i umiejętnościami, a podział taki okazał się działać bardzo skutecznie.

Uważamy, że wykonanie naszej pracy magisterskiej zakończyło się sukcesem. Utworzone przez nas modele szybowców oraz otoczenia wyglądają realistycznie. Wpływ na to ma fakt, że przy rozwiązywaniu konkretnych problemów, które zostały przed nami postawione w trakcie tworzenia modelu posługiwaliśmy się szerokim wachlarzem narzędzi dostarczanych przez niezwykle rozbudowane środowisko *Softimage* w taki sposób, aby efekt końcowy był bogatszy, ciekawszy i zarazem wydajniejszy. Tak więc tworząc modele szybowców, gdzie wymagana była precyzja wykonania oraz dobre odwzorowanie posłużyliśmy się narzędziami manualnego tworzenia geometrii. Stając przed problemem utworzenia terenu, gdzie tym razem nie bazowaliśmy już na konkretnych wzorcach postanowiliśmy utworzyć interaktywny generator przy użyciu *Interactive Creative Environment*, który pozwala w szybkim czasie tworzyć zróżnicowane tereny oraz dostosowywać ich wygląd końcowy do własnych potrzeb przy użyciu szeregu parametrów. Zagadnienie fizyki lotu zostało z kolei rozwiązane przy pomocy skryptu napisanego w języku JScript. Tutaj lepszym wyborem było napisanie własnego programu, oczywiście korzystając z wszelkich udogodnień środowiska ułatwiających połączenie modelu szybowca i otoczenia ze skryptem. Dodatkowo

zaimplementowanie skryptu w postaci operatora skryptowego dodało możliwość sterowania szybowcem w trakcie lotu. Należy zaznaczyć, że przy pisaniu skryptu zrezygnowaliśmy z zastosowania wbudowanych w *Softimage* obiektów jak wiatr, grawitacja czy opór, gdyż ich użycie nie dawałoby nam wystarczającej kontroli nad skomplikowanymi parametrami i niuansami fizyki lotu.

Program *Softimage* okazał się być doskonałym i niezwykle rozbudowanym narzędziem łączącym w sobie cechy środowiska do tworzenia modeli 3D jak i „ożywiania” ich przy pomocy specjalistycznych środowisk programowania.

# 6 Dodatki

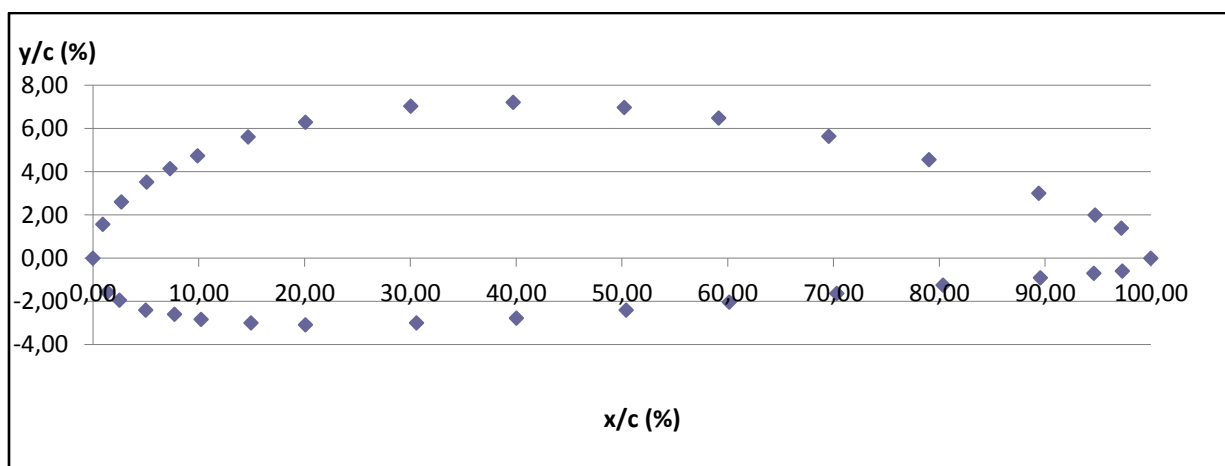
## 6.1 Dodatek A - doświadczenie z profilem skrzydła

W tym rozdziale, przedstawimy w skrócie niektóre wyniki doświadczenia, jakie zostało przeprowadzone w tunelu aerodynamicznym w Instituto Superior Tecnico w Lizbonie. Zostało ono przeprowadzone w ramach przedmiotu Aerodynamika, na który jeden z nas uczęszczał podczas pobytu na stypendium Erasmus. Prowadzącym przedmiot i opiekunem doświadczenia był profesor Luis Eça. Wykonawcami doświadczenia byli André Silverio, Ricardo Mimoso i Łukasz Wędzony.

Celem doświadczenia było jakościowe i ilościowe zbadanie przepływu wokół profilu skrzydła: rozpoznanie efektów występujących dla dużych kątów natarcia oraz wyznaczenie wydajności aerodynamicznej profilu w zależności od wartości kąta natarcia. Umieszczenie skróconego raportu z tego doświadczenia ma na celu wzbogacenie pracy o empiryczne dane które świetnie obrazują i potwierdzają opisaną teorię. Ponadto pokazuje ono wykorzystanie wspomnianych równań i zależności w praktyce, przy doświadczalnym wyznaczaniu oporu i siły nośnej dowolnego profilu.

Doświadczenie zostało przeprowadzone w małym tunelu aerodynamicznym, w którym umieszczone zostało skrzydło. Ponieważ badana była charakterystyka profilu, bez uwzględnienia efektów przy końcówkach skrzydła, na obu krańcach zastosowano prostopadłe do rozpiętości ograniczniki. Skrzydło umieszczone było na osi równoległej do rozpiętości, co umożliwiło zmianę kąta natarcia. Na powierzchni skrzydła umieszczone były czujniki ciśnienia. Rozmieszczenie czujników na profilu przedstawia poniższy Rys. 6.1.





Rys. 6.1 Rozmieszczenie czujników ciśnienia, kształt profilu

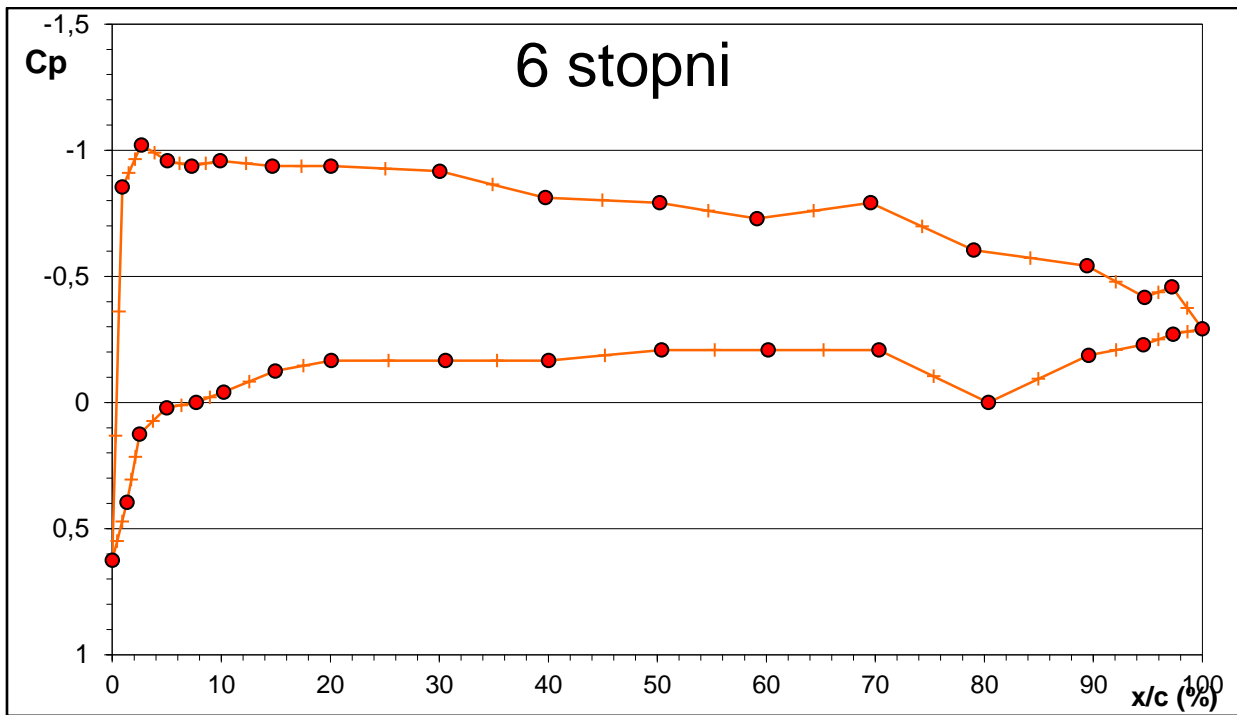
Aby położenie  $x$  i  $y$  sprowadzić do wartości bezwymiarowych stosuje się ich stosunek do cięciwy  $c$ , stąd  $x/c$  to bezwymiarowe położenie wzdłuż cięciwy płata, a  $y/c$  to bezwymiarowe położenie wzdłuż grubości profilu.

Pierwszym zadaniem było sprawdzenie dla jakich kątów natarcia następuje oderwanie linii prądu od powierzchni płata (przecignięcie). Wykonaliśmy to zadanie przytwierdzając do skrzydła bawełniane nitki. Oznaką przecignięcia było ich trzepotanie, w przeciwieństwie do spoczywania na powierzchni skrzydła, co miało miejsce dla małych kątów natarcia.

I tak dla kąta natarcia 16 stopni zauważyliśmy pojedynczą nitkę na górnej części skrzydła przy krawędzi natarcia odstającą od powierzchni. Oznaczało to małe oderwanie linii prądu od powierzchni. Po małym zwiększeniu kąta wszystkie nitki umiejscowione na górze wskazywały na gwałtowne oderwanie na całej górnej powierzchni płata. Takie zjawisko nosi nazwę *Leading Edge Stall*. Dla ujemnych kątów natarcia dotarliśmy do -6 stopni, gdzie oderwanie linii prądu również zaczęło się przy krawędzi natarcia ale na dolnej części skrzydła i postępowało stopniowo wraz ze zmniejszaniem kąta natarcia. Takie zachowanie nosi nazwę *Thin Airfoil Stall*.

Oprócz rozpoznania rodzajów przecignięcia, pierwsza część doświadczenia dała nam również informacje w jakim zakresie kątów natarcia powinniśmy zbierać dane do ilościowej analizy.

Dokonałymi pomiarów ciśnienia na powierzchni płata dla kątów natarcia od -6 stopni do 14 stopni, następnie znając otrzymane wielkości sprowadziliśmy do bezwymiarowego współczynnika ciśnienia  $C_p$ . Dla każdego z pomiarów otrzymaliśmy wykres zależności  $C_p(\alpha)$ . Na Rys. 6.2 znajduje się przykładowy wykres, dla kąta 6 stopni.



Rys. 6.2 Współczynnik ciśnienia  $C_p$  dla kąta natarcia  $6^\circ$   
(oznaczenia jak dla Rys. 6.1)

Obliczenie siły nośnej i oporu wymagało scałkowania współczynnika ciśnienia zgodnie z poniższymi wzorami:

$$C_L = C_Y \cos \alpha - C_X \sin \alpha$$

a) Współczynnik siły nośnej

$$C_D = C_X \cos \alpha + C_Y \sin \alpha$$

b) Współczynnik oporu

Równanie 21 Siła nośna i opór

Gdzie  $C_X$  i  $C_Y$  to odpowiednio składowa x-owa i y-owa ciśnienia, obliczone przez poniższe całkowania:

$$C_Y = - \int_0^1 (C_{pu} - C_{pl}) d\left(\frac{x}{c}\right)$$

a) Y-owa składowa ciśnienia  
 $\frac{yu}{c}$

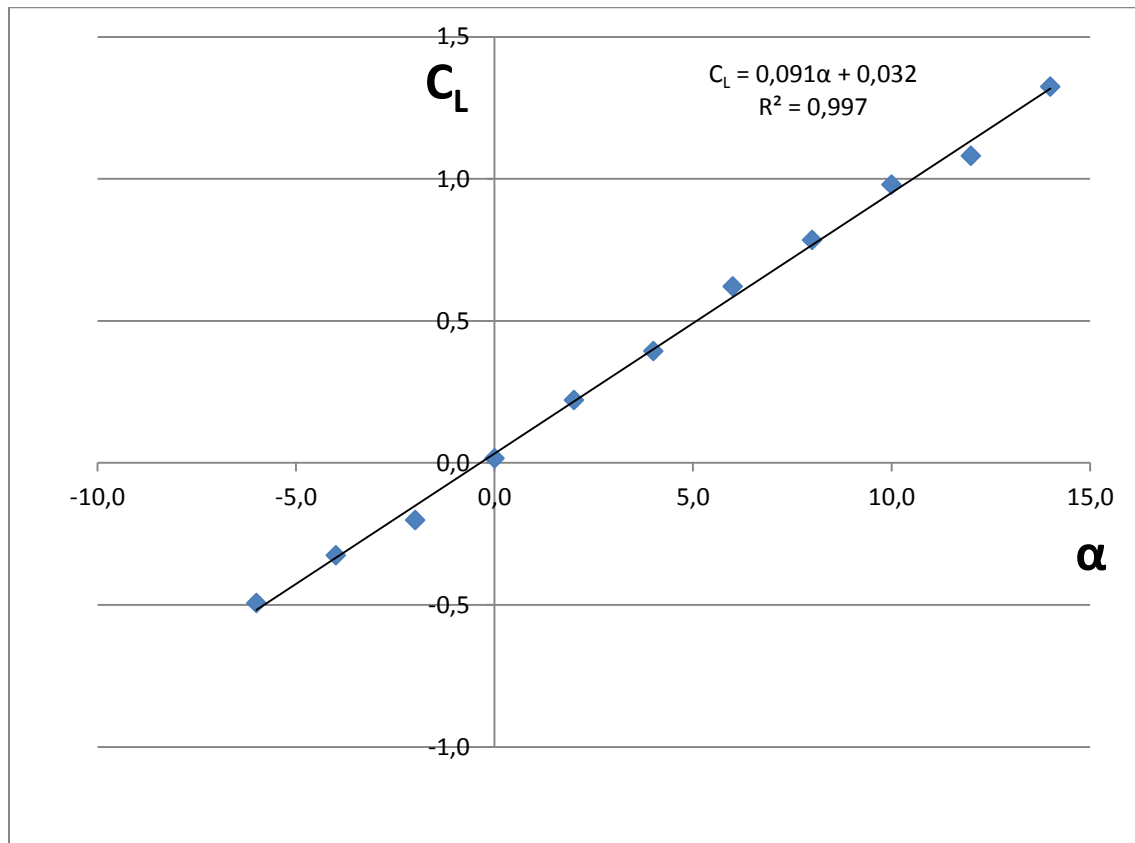
$$C_X = - \int_{\frac{yl}{c}}^{\frac{yl}{c}} (C_{pb} - C_{pa}) d\left(\frac{y}{c}\right)$$

b) X-owa składowa ciśnienia

Równanie 22 Pomocnicze całkowania

Tutaj  $p_u$  i  $p_l$  to indeksy opisujące ciśnienie na górnej i dolnej powierzchni, a  $p_b$  i  $p_a$  oznaczają ciśnienia w punktach przed i po danej pozycji  $y$ .

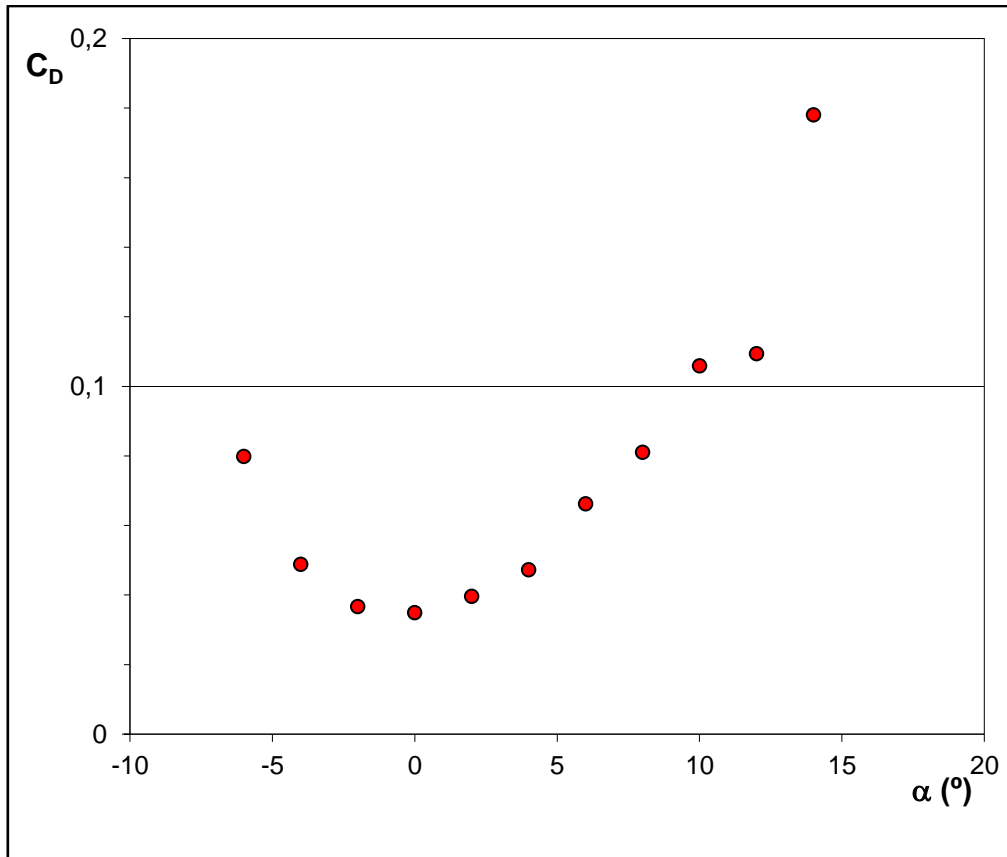
Zestawienie tych wielkości obliczonych dla wszystkich badanych kątów natarcia umożliwia nam następnie przedstawienie poniższych wykresów:



Rys. 6.3 Zależność współczynnika siły nośnej od kąta natarcia

Powyższa ilustracja pokazuje najważniejszą charakterystykę profilu, czyli zależność siły nośnej od zmiany kąta natarcia. Wykres odpowiada zachowaniu profilu dla małych kątów natarcia, a więc wykres jest liniowy.

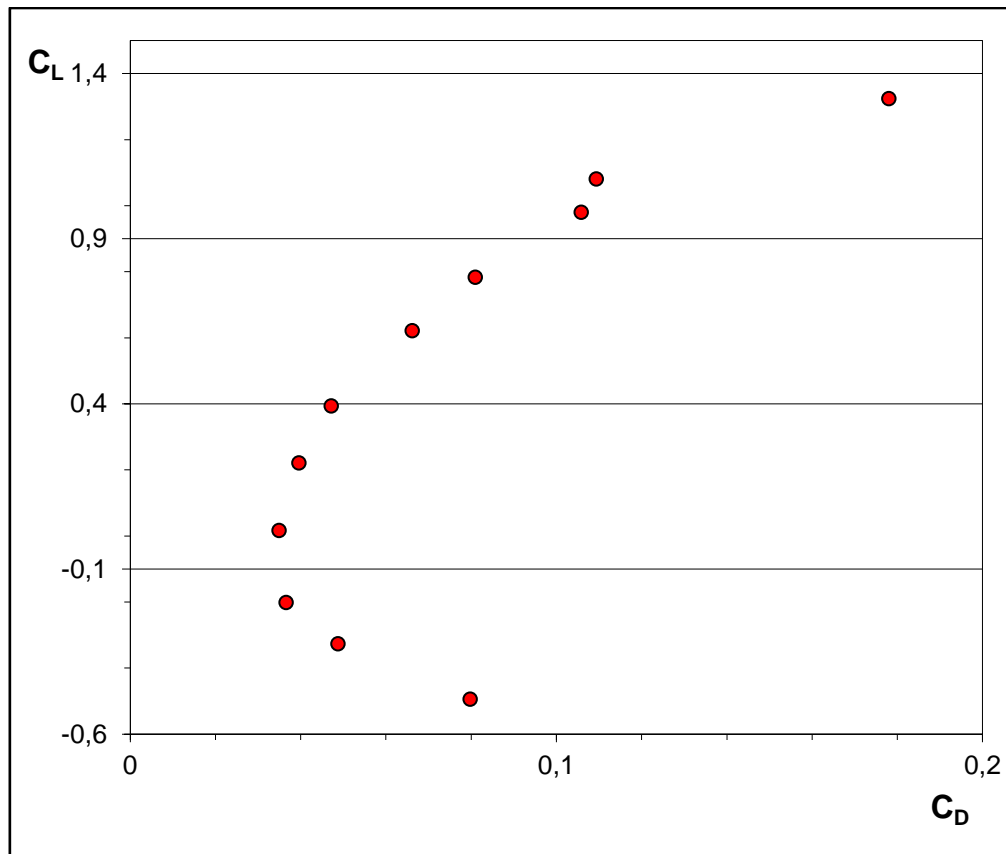
Kolejny wykres przedstawi zachowanie współczynnika oporu.



Rys. 6.4 Zależność współczynnika oporu od kąta natarcia

Kształt tego wykresu wskazuje na to, iż nie możemy tutaj zaniedbać efektów związanych ze skończoną rozpiętością płata.

Ostatnim wykresem pokazującym wydajność skrzydła jest przedstawiona na Rys. 6.5 zależność współczynnika siły nośnej od współczynnika oporu.



Rys. 6.5 Zależność współczynnika siły nośnej od współczynnika oporu

Pokazany powyżej wykres  $C_L(C_D)$  pokazuje nam że wzrost wartości siły nośnej uzyskany przez zwiększenie kąta natarcia zawsze będzie powodował wzrost siły oporu.

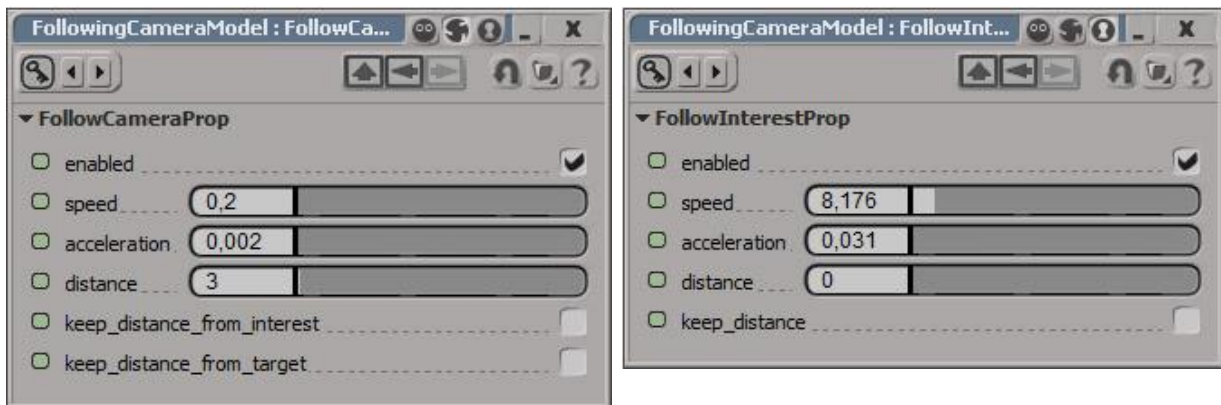
## 6.2 Dodatek B – kamera śledząca

W celu lepszej prezentacji modelu utworzyliśmy specjalną kamerę śledzącą, której można zlecić śledzenie wybranego elementu sceny. Dzięki niej w bardzo prosty sposób możemy tworzyć zarówno dynamiczne ujęcia przelotu szybowca obok kamery, ponieważ ta nigdy nie spuści go z oka, jak i tworzyć dłuższe sekwencje lotu, w których kamera sama podąża za szybowcem zachowując ustawiony dystans i dostosowując się do zmian prędkości, kierunku lotu. Kamera sterowana jest za pomocą operatora skryptowego oraz paru wyrażeń skryptowych i posiada dwa tryby

- powielanie ruch celu
- własne parametry ruchu

W pierwszym trybie działania kamera zwyczajnie powiela ruch śledzonego celu z określonym przez użytkownika opóźnieniem czasowym. W drugim trybie kamera posiada zdecydowanie więcej swobody działania, ponieważ jesteśmy w stanie przypisać jej parametry przyspieszenia, prędkości maksymalnej oraz dystansu minimalnego, który musi zachować od

celu, w ten sposób możemy uzyskać ciekawsze ujęcia w sytuacji, gdy dla przykładu kamera przestaje nadążać za przyspieszającym szybowcem (ponieważ posiada mniejszą szybkość całkowitą), lub potrzebuje więcej czasu, żeby zrównać się z dystansem, który zachowywała przed rozpoczęciem przyspieszania po tym jak prędkość szybowca przestanie wzrastać (mniejsze przyspieszenie). W końcu parametry te możemy modyfikować w trakcie kręcenia ujęć co pozwala robić płynne zbliżenia i oddalenia. W operator skryptowy poza obiektem kamery wyposażony został także jej *point of interest* czyli punkt, na który jest zwrócona. Pozwala to zwiększyć dynamikę ujęć poprzez zaprzestanie kurczowego trzymania się szybowca w centrum kamery.



a) panel sterowania skryptem kamery

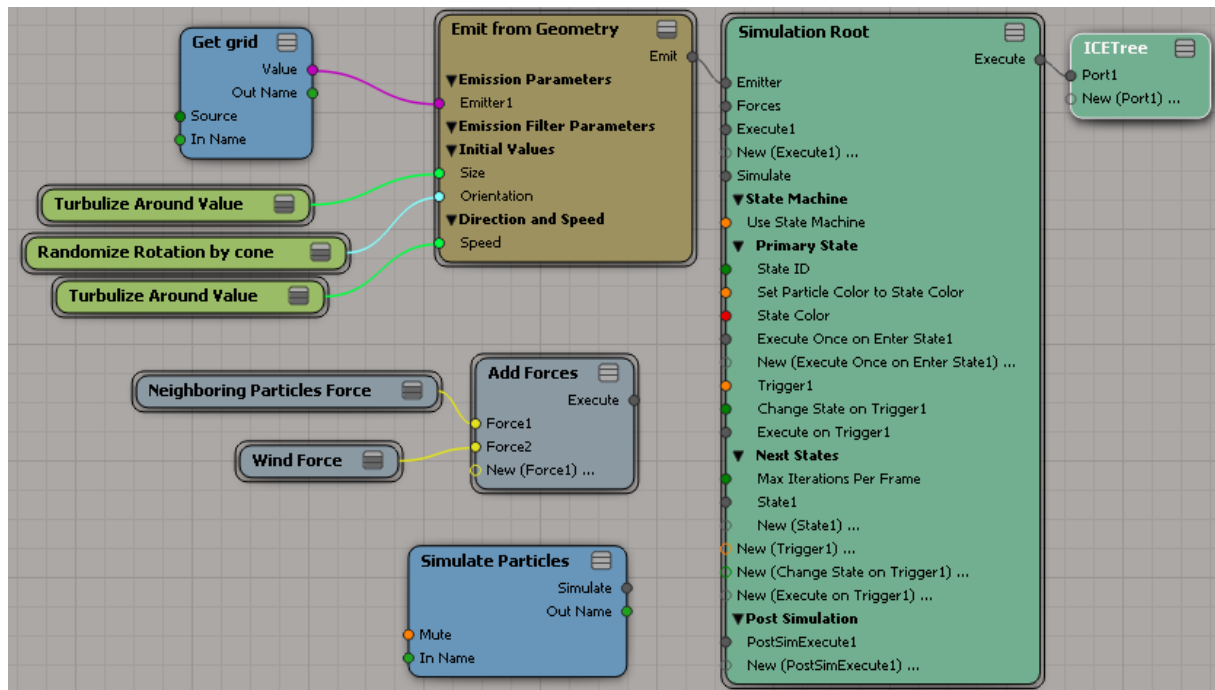
b) panel sterowania skryptem *point of interest* kamery

Rys. 6.6 Panele sterowania skryptem

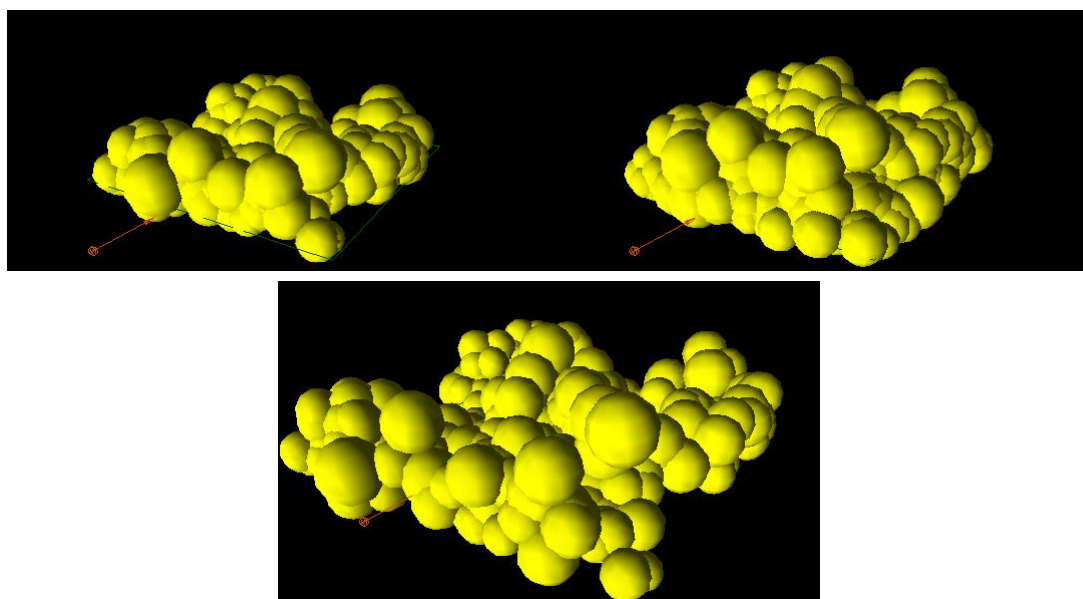
## 6.3 Dodatek C – chmury

Dodatkowym elementem urozmaicającym nasz model otoczenia są chmury. Poza zwiększeniem atrakcyjności wizualnej modelu służą one także do przedstawiania ogólnego kierunku wiatru. Chcąc produkować pełną grę chmur różnych kształtów i rozmiarów stworzyliśmy generator chmur wykorzystując w tym celu środowisko ICE. Podobnie jak przy tworzeniu terenu chmury generować będziemy przy pomocy płaszczyzny z nałożoną teksturą w odcieniach szarości. Tym razem jednak tekstura ta nie posłuży do określenia wartości wypiętrzenia płaszczyzny lecz do określenia jak silnie z danego miejsca płaszczyzny powinny zostać wyemitowane cząsteczki (patrz Particle System) tworzące chmurę. W ten sposób poprzez podłączanie do płaszczyzny różnych tekstur jesteśmy w stanie generować chmury o różnym kształcie (poprzez skalowanie płaszczyzny i zwiększenie ogólnej ilości generowanych cząsteczek zwiększyć możemy również rozmiar bez utraty szczegółowości). Drzewo *ICE Tree* (Rys. 6.7) działa dwustanowo. W pierwszym trybie (podłączona gałąź emitera – tak jak na rysunku) generowane są na płaszczyźnie cząsteczki, z których składać się będzie nasza chmura. Dzięki wpięciu w odpowiednie porty węzłów generujących wartości losowe cząsteczki te będą różniły się rozmiarem, zwrotem (co pozwoli stworzyć iluzję różnicy cząstek

mimo zastosowania tego samego materiału) oraz prędkością. W drugim kroku podłączając węzeł symulacji ustawiony na dodatni kierunek osi Y powodujemy wypiętrzenie chmury. Dodatkowo podłączając opcjonalne siły odpychające od siebie cząsteczki oraz wiatru (szara gałąź) możemy dokonać zmian w strukturze chmury gdy wygląda ona na zbyt zbitą, co może być spowodowane wygenerowaniem w pierwszym kroku zbyt dużej ilości cząsteczek na zbyt małym obszarze). Kolejne kroki symulacji przedstawione zostały na rysunku Rys. 6.8.



Rys. 6.7 Drzewo ICE generujące chmury



Rys. 6.8 Rozwój chmury- kolejno: emisja cząsteczek, symulacja piętrzenia się oraz symulacja z zastosowaniem sił rozpychających

Następnie utworzyliśmy materiał, który dobrze naśladuje chmurę. W tym celu stworzyliśmy materiał objętościowy, który w przeciwieństwie do wykorzystywanych do tej pory materiałów nie działa na płaszczyźnie lecz jest materiałem trójwymiarowym. Materiał ten składa się z obecnych w *Softimage* shaderów generujących objętościowy „szum komórkowy” (*cell scalar*) oraz „szum fraktalny” (*fractal scalar*). Kontrolując parametr gęstości cząsteczek jesteśmy w stanie nadawać chmurom mniej lub bardziej burzowy wygląd ( Rys. 6.9 ).



a) chmura w trakcie pogodnego dnia



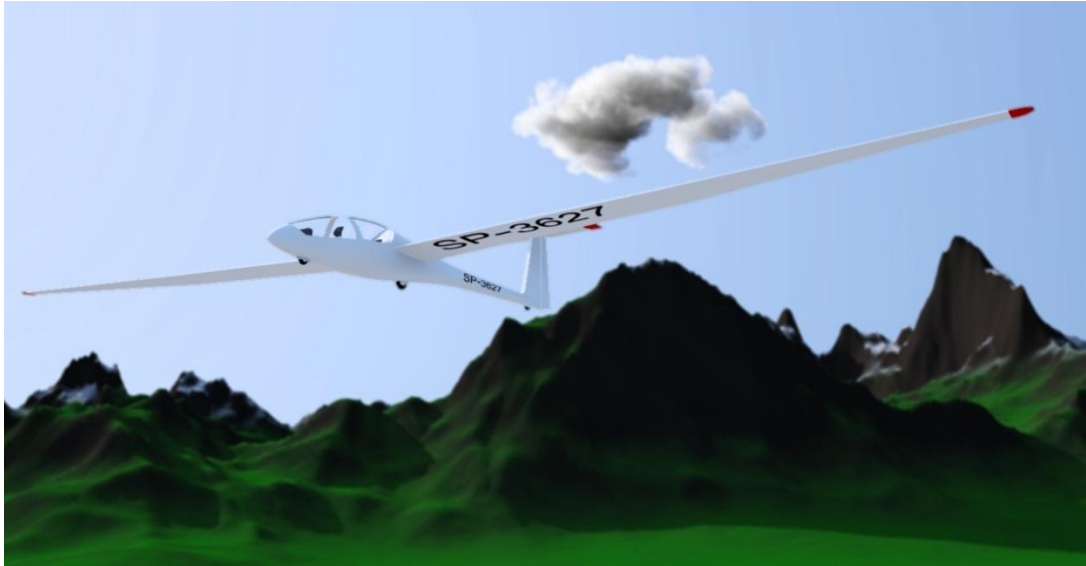
a) chmura burzowa

Rys. 6.9 Efekt końcowy generatora chmur



## Particle System

Pod pojęciem tym określa się technikę grafiki 3D polegającą na tworzeniu na scenie obiektów składających się z dużej ilości prostych elementów. Elementy te można następnie renderować przy użyciu tekstur 3D w celu uzyskania efektów takich jak chociażby dym, ogień, śnieg czy też tak jak w naszym przypadku chmury. Uzyskuje się to poprzez odpowiednie emitowanie cząsteczek z określonego źródła, a następnie poddawanie ich określonym wzorcom takim jak choćby nadawanie im własności fizycznych (ciężar, prędkość unoszenia, czynniki zewnętrzne jak wiatr, turbulencje)



# 7 Spis ilustracji

Rys. 1.1 Plany maszyny latającej Leonarda Da Vinci .....	6
Rys. 1.2 Replika szybowca Georga Cayley'a w trakcie lotu (rok 1973) .....	7
Rys. 2.1 Siły działające na szybowiec.....	10
Rys. 2.2 Przykładowe ciało poruszające się w ziemskiej atmosferze .....	12
Rys. 2.3 Przykładowy profil lotniczy poruszający się w powietrzu.....	15
Rys. 2.4 Schematyczne przedstawienie przepływu laminarnego oraz turbulentnego .....	16
Rys. 2.5 Demonstracja praktycznego zastosowania równania Bernoulliego.....	17
Rys. 2.6 Przykładowy profil skrzydła w ruchu .....	19
Rys. 2.7 Przykłady efektów wynikających z różnicy ciśnień płynu pozostającego w spoczynku i ruchomego, w których możemy zaobserwować tzw. paradoks hydrodynamiczny.....	19
Rys. 2.8 Rozkład prędkości strug powietrza w warstwie granicznej .....	20
Rys. 2.9 Porównanie liniowej i turbulentnej warstwy granicznej.....	22
Rys. 2.10 Przejście opływu liniowego w turbulentny na skrzydle szybowca .....	23
Rys. 2.11 porównanie kąta oderwania i rozmiaru nurtu .....	25
Rys. 2.12 Wyprowadzenie współczynnika lepkości.....	26
Rys. 2.13 Wiry powstające na końcówkach skrzydeł .....	27
Rys. 2.14 Wpływ zwiększenia wydłużenia skrzydła na istotność poprawki na indukowaną siłę nośną .....	29
Rys. 2.15 Tylni spoiler auta F1 zakończony wingletami .....	30
Rys. 2.16 Skrzydło samolotu zakończone wingletami.....	30
Rys. 2.17 Supermarine Spitfire ze skrzydłami w kształcie elipsy.....	30
Rys. 2.18 Samolot ze skrzydłami w kształcie pierścienia dookoła kadłuba.....	31
Rys. 2.19 Rodzaje skosów skrzydeł.....	32
Rys. 2.20 Przykład powierzchni nośnej – skrzydło szybowca.....	33
Rys. 2.21 Przykłady powierzchni nośnych .....	34
Rys. 2.22 Nomenklatura profilu lotniczego .....	35
Rys. 2.23 Efekt przeciągnięcia .....	36
Rys. 2.24 Siła nośna jako funkcja kąta natarcia.....	37
Rys. 2.25 Środki parcia prostych powierzchni nośnych w zależności od kąta natarcia .....	38
Rys. 2.26 Przykłady profili powierzchni nośnych .....	39

Rys. 2.27 Profil NACA 0009.....	40
Rys. 2.28 Profil NACA 4412.....	41
Rys. 2.29 Profil NACA 23015.....	42
Rys. 2.30 Profil NACA 23021.....	42
Rys. 2.31 Rozkład ciśnienia na powierzchni profilu liniowego.....	43
Rys. 2.32 Porównanie przepływu wokół zwykłego i liniowego profilu.....	43
Rys. 2.33 Profil liniowy NACA 66-209 ( $a=1$ ).....	44
Rys. 2.34 Zależność $C_L$ ( $C_D$ ) dla profilu liniowego (NACA 63-009), tradycyjnego (NACA 0009) oraz tradycyjnego z otwartymi klapami (0009 with split flap).....	45
Rys. 2.35 Profil FX 73-K-170/22.....	46
Rys. 2.36 Zależność współczynnika siły nośnej $C_L$ od kąta natarcia $\alpha$ .....	46
Rys. 2.37 Zależność współczynnika siły nośnej $C_d$ od kąta natarcia $\alpha$ .....	47
Rys. 2.38 Zależność wartości współczynnika siły nośnej $C_L$ wartości współczynnika oporu $C_d$ .....	47
Rys. 2.39 Elementy mechanizacji skrzydła:.....	48
Rys. 2.40 Zasada działania slotu szczelinowego.....	49
Rys. 2.41 Wpływ klap na powstawanie siły nośnej na płaskiej powierzchni dla zerowego kąta natarcia.....	49
Rys. 2.42 Przykładowy efekt użycia klap.....	50
Rys. 2.43 Niektóre rodzaje klap.....	51
Rys. 2.44 Rodzaje usterzenia szybowca.....	52
Rys. 2.45 Schematyczny rysunek szybowca z wysuniętymi hamulcami aerodynamicznymi... ..	53
Rys. 2.46 Przykłady spojlerów.....	54
Rys. 3.1 Plany szybowca <i>Glaser-Dirks DG-500</i> .....	56
Rys. 3.2 Rotoskop modelu w środowisku <i>Softimage</i> .....	57
Rys. 3.3 Tworzenie okręgów oraz zastosowanie funkcji <i>loft</i> .....	57
Rys. 3.4 Krzywa Béziera w programie <i>Softimage</i> .....	58
Rys. 3.5 Porównanie obiektu bazowego i utworzonej na jego bazie powierzchni podziału ... ..	59
Rys. 3.6 Usuwanie nadmiarowej geometrii z okolic ogona modelu.....	59
Rys. 3.7 Modelowanie szyb.....	59
Rys. 3.8 Geometria usterzenia.....	60
Rys. 3.9 Zastosowanie podziału Catmull-Clark.....	60
Rys. 3.10 Krzywa Béziera naśladowująca kształt profilu skrzydła.....	61
Rys. 3.11 Utworzone krzywe profilu skrzydła.....	62
Rys. 3.12 Utworzenie skrzydła z posiadanych krzywych oraz zwiększenie rozdzielczości elementu.....	62
Rys. 3.13 Usuwanie nadmiarowej geometrii ze skrzydła. Krawędzie do usunięcia zostały zaznaczone na czerwono.....	63
Rys. 3.14 Wyodrębnianie z elementu skrzydła geometrii hamulca.....	63
Rys. 3.15 Proces tworzenia lotek skrzydła.....	63
Rys. 3.16 Wypełnienie wnętrza kokpitu.....	64
Rys. 3.17 Tworzenie powierzchni przy użyciu funkcji <i>loft</i> .....	65

Rys. 3.18 Zastosowanie <i>Add Edge Loop Tool</i> .....	65
Rys. 3.19 Tworzenie ścian przy użyciu narzędzia <i>Add/Edit Polygon Tool</i> .....	65
Rys. 3.20 Opcje presetu <i>Physical sun and sky</i> .....	66
Rys. 3.21 Wygląd szybowca w utworzonym środowisku .....	67
Rys. 3.22 Rysunek obrazujący schematyczne działanie <i>Final Gathering</i> .....	68
Rys. 3.23 Efekt zastosowania materiałów .....	69
Rys. 3.24 Koła oraz znakowania – „numery rejestracyjne” i czerwone pasy poprawiające widoczność szybowca.....	70
Rys. 3.25 Wnętrze szybowca .....	71
Rys. 3.26 Model Schweizer SGS 2-32 .....	72
Rys. 3.27 Model szybowca <i>Fafnir</i> .....	74
Rys. 3.28 Przykładowe drzewo <i>ICE</i> powodujące losową zmianę położenia wierzchołków obiektu (patrz <i>ICE</i> - przykład). .....	75
Rys. 3.29 Typy danych środowiska <i>ICE</i> .....	77
Rys. 3.30 Niektóre węzły konwersji typów <i>ICE</i> .....	77
Rys. 3.31 Efekt działania skonstruowanego drzewa na płaszczyźnie.....	78
Rys. 3.32 Polimorfizm na przykładzie węzła <i>Multiply</i> , który posiada możliwość mnożenia zarówno wartości <i>Integer</i> , <i>Scalar</i> , <i>Vector 3D</i> itp.....	79
Rys. 3.33 płaszczyzna z teksturą w postaci heightmapy .....	79
Rys. 3.34 Przykładowa heightmapa.....	80
Rys. 3.35 Wartości kolorów oraz odpowiadające im wartości parametru <i>Lightness</i> .....	80
Rys. 3.36 Zmiana wysokości położenia wierzchołka w zależności od wartości <i>Lightness</i> i parametru skali.....	81
Rys. 3.37 Wygląd generowanego terenu w zależności od gęstości siatki obiektu .....	81
Rys. 3.38 Wektory normalne powierzchni dla poszczególnych wierzchołków .....	82
Rys. 3.39 Tworzenie map wagowych w zależności od wektorów normalnych powierzchni... ..	82
Rys. 3.40 Materiał przypisany tworzonemu terenowi .....	83
Rys. 3.41 Wpływ mnożnika wag koloru skał na wygląd terenu .....	84
Rys. 3.42 Materiał z uwzględnieniem koloru śniegu .....	85
Rys. 3.43 Teren ze śniegiem .....	85
Rys. 3.44 Zmiana parametru wtapienia śniegu .....	86
Rys. 3.45 Zmiana wyglądu terenu w zależności od wysokości pokrywy śniegu, dla parametru wtapienia 20% .....	87
Rys. 4.1 Kąty Eulera .....	91
Rys. 4.2 Podział szybowca .....	92
Rys. 4.3 Okno edytora skryptów .....	95
Rys. 4.4 Przykład zastosowania obu modeli programistycznych .....	95
Rys. 4.5 Właściwości kinematyczne obiektu typu <i>null</i> .....	97
Rys. 4.6 Przygotowanie sceny .....	98
Rys. 4.7 Stosowanie edytora wyrażeń.....	98
Rys. 4.8 Zmiana parametru <i>distance</i> w zależności od odległości kamery od obiektu .....	99

Rys. 4.9 Efekt końcowy .....	99
Rys. 4.10 Okno edytora operatora skryptowego .....	100
Rys. 4.11 Połączenia operatora skryptowego .....	103
Rys. 4.12 Panel sterowania szybowcem.....	103
Rys. 4.13 Inicjalizacja tablicy elementów konstrukcyjnych szybowca .....	104
Rys. 4.14 Obliczenie elementów macierzy tensora momentu bezwładności.....	104
Rys. 4.15 Wpływ słońca na występujące prądy powietrzne .....	105
Rys. 4.16 Sterowanie wiatrem .....	106
Rys. 4.15 Funkcja o nazwie krokCalosci .....	107
Rys. 4.16 Funkcja o nazwie krokElementu .....	107
Rys. 6.1 Rozmieszczenie czujników ciśnienia, kształt profilu .....	113
Rys. 6.2 Współczynnik ciśnienia $C_p$ dla kąta natarcia $6^\circ$ .....	114
Rys. 6.3 Zależność współczynnika siły nośnej od kąta natarcia .....	115
Rys. 6.4 Zależność współczynnika oporu od kąta natarcia .....	116
Rys. 6.5 Zależność współczynnika siły nośnej od współczynnika oporu .....	117
Rys. 6.6 Panele sterowania skryptem .....	118
Rys. 6.7 Drzewo <i>ICE</i> generujące chmury .....	119
Rys. 6.8 Rozwój chmury- kolejno: emisja cząsteczek, symulacja piętrzenia się oraz symulacja z zastosowaniem sił rozpychających.....	119
Rys. 6.9 Efekt końcowy generatora chmur .....	120

# 8 Bibliografia



Antoni Adamczyk. Wykłady i Animacje z Fizyki Ogólnej. 2008.  
<<http://www.if.pw.edu.pl/~bibliot/archiwum/adamczyk/WykladyFO.html>>.

Autodesk. Softimage User's Guide. 2012. <[www.autodesk.com/softimage-userguide](http://www.autodesk.com/softimage-userguide)>.

David M. Bourg. Fizyka dla programistów gier. Gliwice: Helion, 2003.

Federal Aviation Administration U.S. Department of Transportation. Pilot's Handbook of Aeronautical Knowledge. 2008.

Frank M. White. Fluid Mechanics. McGraw - Hill, 2003.

Louis Eca. Opublikowane studentom materiały z wykładów z Aerodynamiki. Lizbona: Instituto Superior Tecnico, 2011.

Martin Simons. Sailplanes 1920 - 1945. EQIP, 2006.



— . Sailplanes 1945 - 1965. EQIP, 2006.

— . Sailplanes 1965 - 2000. EQIP, 2004.

NASA Glenn Research Center. Glenn Learning Technologies Project. 2008.  
<<http://www.grc.nasa.gov/WWW/k-12/>>.

Noel Kruse. Fly Better, Book One (Second Edition), Aerodynamics and other Stuff. 2010.