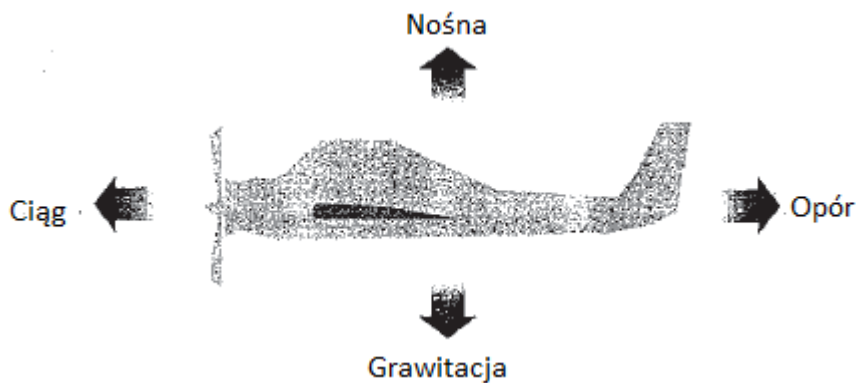
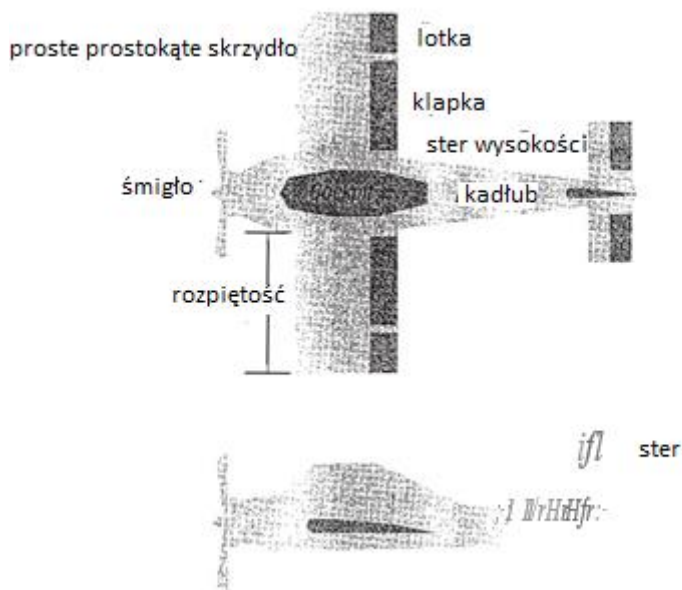


XV. Samolot

Jeśli masz zamiar napisać grę symulacyjną lotu, jednym z najważniejszych aspektów silnika gry będzie model Twojego samolotu. Tak, twoja grafika 3D, interfejs użytkownika, historia, symulacja systemu awioniki i kodowanie są ważne, ale to, co naprawdę definiuje zachowanie samolotu, który symulujesz, to twój model lotu. Zasadniczo jest to twoja uproszczona wersja fizyki lotu samolotu, czyli twoje założenia, twoje przybliżenia i wszystkie formuły, których użyjesz do obliczenia masy, bezwładności oraz siły i momentów unoszenia i przeciągania. Istnieją cztery główne siły działające na samolot podczas lotu: grawitacja, siła ciągu, siła nośna i siła uciążu. Grawitacja jest oczywiście siłą, która ma tendencję do ciągnięcia samolotu na ziemię, podczas gdy siła nośna generowana jest przez skrzydła (lub powierzchnie ratunkowe) samolotu, aby przeciwdziałać grawitacji i umożliwić jej utrzymanie w powietrzu. Nacisk generowany przez sprężynę napędową samolotu (silnik odrzutowy lub śmigło) zwiększa prędkość statku powietrznego i umożliwia podniesienie powierzchni. Na koniec, przeciągnięcie przeciwdziała sile naporu, która ma tendencję do hamowania płynności statku powietrznego. Rysunek 1 ilustruje te siły.



Już wcześniej omawiałem siłę grawitacji we wcześniejszych rozdziałach, więc nie odniosę się do tego ponownie w tym czcicielu, z wyjątkiem tego, że patrząc na wszystkie siły działające na samolot, rumpla wszystkich sił windowych musi być większa niż lub równy sile grawitacji, jeżeli statek powietrzny ma utrzymać równowagę. Aby odnieść się do pozostałych trzech sił działających na statku powietrznym, powiem uproszczony, ogólny model samolotu i wykorzystam go jako iluzoryczny przykład. Istnieje zbyt wiele samolotów i konfiguracji, aby traktować je wszystkie w tym krótkim rozdziale. Ponadto przedmiot aerodynamiki jest bardzo szeroki i złożony. Dlatego model, który obejrzymy, będzie miał typową konfigurację poddźwiękową, jak pokazano na rysunku 2



W tej konfiguracji główne powierzchnie podnoszące (duże skrzydła) są umieszczone w samolocie z relatywnie mniejszymi powierzchniami podnoszącymi w kierunku ogona. Jest to podstawowy układ większości istniejących samolotów. Będę musiał dokonać dalszych uproszczających założeń, dzięki którym nawet ten uproszczony model będzie mógł być zarządzany. Ponadto opieram się na danych empirycznych i formułach do obliczania siły unoszenia i przeciągania.

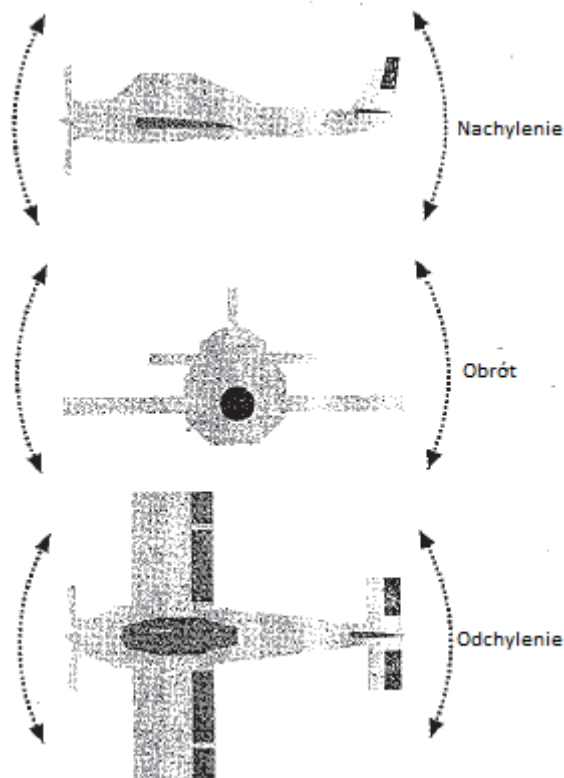
Geometria

Geometria

Zanim przejdziemy do nośnej, ciągu i oporu, musimy przejść przez jakąś podstawową geometrię i warunki, aby upewnić się, że mówimy tym samym językiem. Znajomość tych terminów pomoże Ci szybko znaleźć to, czego szukasz, przeszukując referencje, które dostarczę później. Najpierw spójrz na układ naszego modelu samolotu na rysunku. Główna część samolotu, część zwykle zajmowana przez ładunek i ludzi, nazywana jest kadłubem. Skrzydła są dużymi prostokątnymi powierzchniami podnoszącymi wystającymi z kadłuba w pobliżu przedniego końca. Dłuższy wymiar skrzydła nazywany jest jego rozpiętością, a jego krótszy wymiar nazywany jest długością cięciwy lub po prostu akordem. Stosunek kwadratu do kwadratu nazywany jest współczynnikiem kształtu, a dla prostokątnych skrzydeł zmniejsza się do proporcji przęsła do akordu. W naszym modelu lotki znajdują się na zewnętrznych końcach skrzydeł. Klapy znajdują się również na skrzydłach pokładów lotek. Małe, podobne do skrzydeł, powierzchnie. Ucho ogona nazywa się elatorami. A pionowa klapa znajdująca się na rufowym końcu ogona to ster. Powiem więcej o tym, co te powierzchnie kontrolne robią później. Przyjrzenie się przekrojowi poprzecznemu skrzydła, jak pokazano na rysunku 3, pomaga zdefiniować kilka dodatkowych terminów.

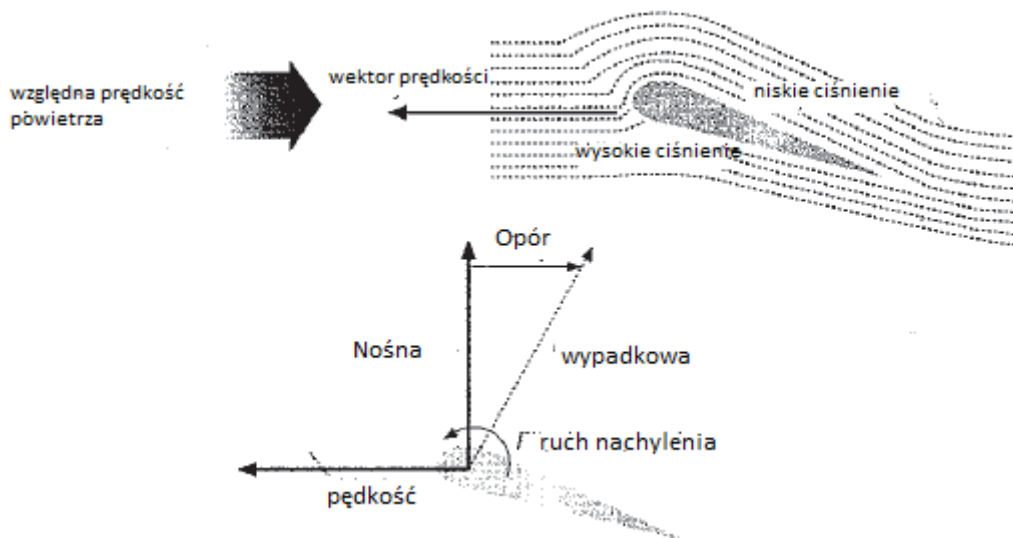


Płat pokazany na rysunku 3 jest typowym profilowanym płatem. Wygięcie przedstawia krzywiznę płata. Jeśli narysujesz linię prostą od krawędzi spływu do krawędzi wiodącej, skończysz z tak zwaną linią akordów. Naw, jeśli podzielisz profil na kilka przekrojów, jak plasterki w bochenku chleba, przechodząc od krawędzi spływu do krawędzi natarcia, a następnie narysuj zakrzywioną linię przechodzącą przez środek każdej grubości przekroju, kończysz z średnią linią wygięcia. Maksymalna różnica pomiędzy średnią linią wygięcia a linią akordów jest miarą pochylenia płata. Kąt mierzony między kierunkiem przemieszczania się płata (wektor prędkości względnej płata podczas jego lotu przez powietrze), a linią akordów nazywany jest bezwzględnym kątem natarcia. Gdy samolot jest w locie, może obracać się wokół dowolnej osi. Standardową praktyką jest zawsze odnosić się do obrotów powietrznych około trzech osi w stosunku do pilota. W ten sposób osie te są przymocowane do przestrzeni powietrznej, że tak powiem, niezależnie od jej rzeczywistej orientacji w przestrzeni trójwymiarowej; są to oś nachylenia, oś obrotu i oś odchylenia. Oś obrotu biegnie poprzecznie w poprzek statku powietrznego, czyli w kierunku lewej burty. Skok obrotu jest wtedy, gdy nos samolotu jest podnoszony lub opuszczany z perspektywy pilota. Oś rolki biegnie wzdłużnie przez centrum samolotu. Ruchy rolki (obroty) wokół tej osi skutkują podniesieniem lub obniżeniem skrzydeł po dowolnej stronie pilota. Ostatecznie oś odchylenia jest osią pionową, wokół której nosek poduszek powietrznych obraca się w kierunku od lewej do prawej (lub od prawej do lewej) względem pilota. Te obroty są zilustrowane na rysunku 4



Nośna i Opór

Kiedy płat przemieszcza się przez płyn taki jak powietrze, powstaje siła nośna. Mechanizmy, za pomocą których to występuje, są podobne do tych w przypadku siły oddziaływania Magnusa omawianej w poprzednim rozdziale, w której wciąż obowiązuje prawo Bernoulliego. Jednak tym razem, zamiast obracania, kształt i kąt natarcia płata wpływają na przepływ powietrza, tworząc podnośnik. Rysunek 5 pokazuje sekcję płata poruszającą się w powietrzu z prędkością V

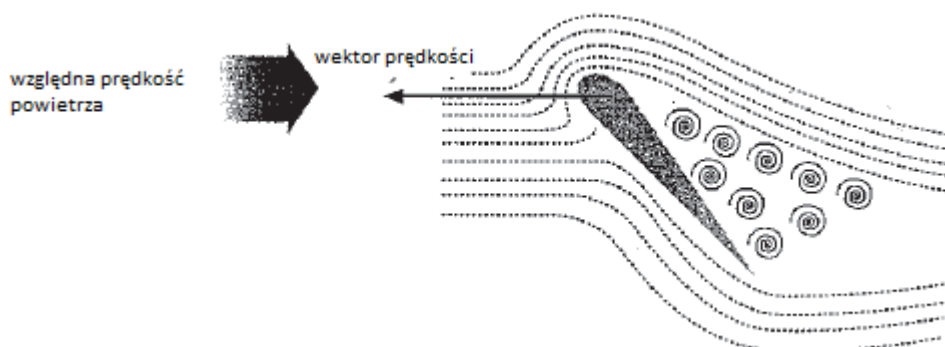


V to względna prędkość pomiędzy płatem a niezakłóconym powietrzem przed płatem. Gdy powietrze uderza i porusza się wokół folii, rozszczepia się w przednim punkcie stagnacji usytuowanym w pobliżu krawędzi natarcia folii, tak że powietrze przepływa zarówno pod, jak i nad folią. Powietrze przepływające pod folią zostaje odchylone w dół, podczas gdy powietrze przepływające nad folią przyspiesza, gdy przechodzi wokół krawędzi natarcia i ponad powierzchnią powietrza, które następnie płynnie przepływa od krawędzi spływu; jest to tak zwany warunek Kutta. Idealnie, warstwa graniczna pozostaje "przymocowana" do folii bez oddzielania, jak w przypadku kuli omawianej w poprzedniej części. Stosunkowo szybko poruszające się powietrze nad płatem powoduje obszar niskiego ciśnienia nad folią (pamiętaj, że równanie Bernoulliego pokazujące ciśnienie jest odwrotnie proporcjonalne do prędkości przepływu płynu). Powietrze uderzające i poruszające się pod spodem płata tworzy obszar o stosunkowo wysokim ciśnieniu. Połączonym efektem tego schematu przepływu jest utworzenie obszarów o stosunkowo niskim i wysokim ciśnieniu powyżej i poniżej płata. To właśnie ta różnica ciśnień powoduje wzrost siły nośnej. Z definicji siła nośna jest prostopadła do linii lotu, to znaczy wektora prędkości. Zwróć uwagę, że płat nie musi być wygięty, aby wygenerować siłę nośną; płaska płyta zorientowana pod kątem natarcia w stosunku do przepływu powietrza również generuje siłę nośną. Podobnie, płat powietrzny nie musi mieć kąta natarcia. Profilowane profile mogą generować wzrost przy zerowych, a nawet ujemnych kątach natarcia. Tak więc, ogólnie, całkowita siła nośna na płacie składa się z dwóch elementów: podnośnika z powodu pochylenia i uniesienia z powodu kąta natarcia. Teoretycznie grubość płata nie przyczynia się do podniesienia. W końcu możesz mieć cienkie zakrzywione skrzydło, tak jak w przypadku skrzydeł wykonanych z tkanin takich jak te używane na lotnie. W praktyce grubość jest wykorzystywana ze względów konstrukcyjnych. Co więcej, grubość przy krawędzi natarcia może pomóc opóźnić przeciągnięcie (więcej o tym za chwilę). Różnica ciśnień pomiędzy górną i dolną powierzchnią płata powoduje również siłę oporu, która działa zgodnie z przeciwną do wektora prędkości. Siły podnoszenia i oporu są prostopadłe do siebie i leżą w płaszczyźnie. Ukazany przez wektor prędkości i wektor normalny (prostopadły) do linii akordów płata. Te dwa składniki siły, unoszenie i opadanie, po połączeniu dają wypadkową siłę działającą na płat w locie. Zilustrowano to na rysunku 5. Zarówno podnoszenie, jak i przeciąganie są funkcjami gęstości powietrza, prędkości, lepkości, pola powierzchni, współczynnika kształtu, \cdot i kąta natarcia. Tradycyjnie, właściwości podnoszenia i przeciągania danego wzoru folii wyrażane są w postaci współczynników nienośnych:

$$C_L = L / [(1/2)\rho V^2 S]$$

$$C_D = D / [(1/2)\rho V^2 S]$$

gdzie S jest powierzchnią planistyczną skrzydła (rozpiętość akordów dla skrzydeł prostokątnych), L jest siłą nośną, D jest siłą oporu, V jest prędkością w powietrzu, a ρ jest gęstością powietrza. Współczynniki te są eksperymentalnie określone na podstawie testów w tunelach aerodynamicznych modeli płatów pod różnymi kątami ataku. Wyniki tych testów są zwykle przedstawiane jako wykresy współczynnika wzrostu i oporu w funkcji kąta natarcia. Najbardziej znaną rodziną wzorów profili płatów i danych testowych są sekcje profili NACA. „Theory of Wing Sections” autorstwa Iry H. Abbotta i Alberta E. Von Doenhoffa zawiera bogactwo danych dotyczących wyciągów i opadów dla praktycznych projektów płatów. W praktyce, przepływ powietrza wokół skrzydła nie jest sztywny dwuwymiarowo, to znaczy, przepływa równomiernie nad każdym równoległym przekrojem skrzydła, i tam przepływa przez niego spontanicznie przepływ powietrza wzdłuż skrzydła. Mówi się, że przepływ jest trójwymiarowy. Im bardziej trójwymiarowy jest przepływ, tym mniej efektywne skrzydło. Ten efekt jest redukowany na długich skrzydłach o wysokim współczynniku kształtu (i skrzydłach z płytkami końcowymi, gdzie zwiększa się efektywny współczynnik kształtu); w ten sposób skrzydła o wysokim współczynniku proporcji są stosunkowo bardziej wydajne. Aby uwzględnić wpływ współczynnika kształtu, sekcje boczne o różnych współczynnikach kształtu dla danej konstrukcji folii są zwykle testowane w taki sposób, aby wytworzyć rodzinę krzywych wytrzymałości i oporu w funkcji kąta natarcia. Istnieją inne czynniki geometryczne, które wpływają na przepływ wokół skrzydeł. Wracając do figury 6, zauważysz, że współczynnik oporu rośnie gwałtownie wraz z kątem natarcia. Jest to uzasadnione, ponieważ można oczekiwać zucia, aby wytworzyć największą siłę oporu, gdy zorientuje się płasko na lub prostopadle do strumienia powietrza. Spojrzenie na krzywą współczynnika wzrostu, która początkowo wzrasta liniowo wraz z kątem natarcia, pokazuje, że przy pewnym kącie natarcia współczynnik wzrostu osiąga wartość maksymalną. Ten kąt nazywany jest krytycznym kątem natarcia. W przypadku kątów wykraczających poza wartość krytyczną współczynnik nośności szybko spada, a płat (lub skrzydło) zatrzymuje się i przestaje wytwarzać siłę nośną. To jest złe. Gdy samolot utknie w powietrzu, zacznie gwałtownie spadać, aż pilot skoryguje sytuację przeciągnięcia, na przykład zmniejszając skok i zwiększając ciąg. Kiedy następuje przeciągnięcie, powietrze nie przepływa już gładko ponad krawędzią spływu, a odpowiadający wysoki kąt natarcia pojawia się w separacji przepływu, jak pokazano na rysunku 7.



Utracie podnośnika towarzyszy również wzrost oporu. Teoretycznie siła wypadkowa działająca na profil działa przez punkt znajdujący się na jednej czwartej długości cięciwy za krawędzią prowadzącą. Nazywa się to ćwierć-akordowym punktem. W rzeczywistości wypadkowa linia działania siły będzie się różnić w zależności od kąta ataku, rozkładu ciśnienia i prędkości, a także innych czynników. Jednak w praktyce uzasadnione jest założenie, że linia działania przebiega przez punkt kwadraturowy dla typowych warunków operacyjnych. Aby uwzględnić różnicę między faktyczną linią działania

wypadkowej a ćwierćnutowego punktu, należy wziąć pod uwagę moment pochylenia wokół punktu akordowego w ćwierć łatac. Ten moment nachylenia zwykle ma tendencję do przechylania krawędzi natarcia płata w dół. W niektórych przypadkach ta chwila jest stosunkowo niewielka w porównaniu do innych momentów działających na samolocie i może być zaniedbana. Wyjątkiem może być sytuacja, gdy folia ma odchylone klapy. Klapy są urządzeniami kontrolnymi, które służą do zmiany kształtu folii, aby zmienić jej charakterystykę podnoszenia. Rysunek 6 pokazuje również typowe współczynniki nośności, oporu i momentu dla profilu wyposażonego w gładką klapę odchyloną w dół o 15 stopni. Zwróć uwagę na znaczny wzrost momentu podnoszenia, przeciągania i pochylenia, gdy klapa jest odchylona. Teoria Sekcji Skrzydłowych dostarcza również dane dla płatków skrzydłowych dla kątów klap między -15 a 60 stopni.

Inne siły

Najbardziej znaną siłą, którą muszę jeszcze omówić, jest siła napędowa. Pchnięcie zapewnia ruch do przodu, a bez niego skrzydła samolotu nie mogą generować podnośnika, a samoloty nie będą latać. Ciąg, generowany przez śmigło lub silnik odrzutowy, jest zwykle wyrażany w funtach, a wspólny stosunek stosowany do porównania względnych zalet zasilania samolotu to stosunek ciągu do masy. Stosunek ciągu do masy jest maksymalnym ciągiem dostarczanym przez instalację napędową podzieloną przez całkowitą wagę statku powietrznego. Gdy stosunek ciągu do masy jest większy niż 1, samolot jest zdolny do pokonania grawitacji podczas pionowego wznoszenia. W tym przypadku winda generowana przez skrzydła nie pomaga w utrzymaniu (lub zwiększaniu) wysokości; jednakże podnoszenie jest generowane przez stil, co ma tendencję do odciągania samolotu od pionowej trajektorii. Ciąg, niezależnie od tego, czy jest wytwarzany przez śmigło, czy silnik odrzutowy, jest funkcją gęstości powietrza. Na dużych wysokościach zmniejsza się gęstość powietrza (i zawartość tlenu), a siła ciągu odpowiednio się zmniejsza. W pewnym momencie silniki zatrzymają się i przestaną napędzać samolot wyżej. Jeśli byłeś kiedyś na pokazie lotniczym, prawdopodobnie widziałeś ten rodzaj kaskaderskiego wyczynu celowo. Oprócz grawitacji, siły ciągu i uniesienia skrzydła i oporu, istnieją inne siły działające na samolot w locie. Są to siły oporu (i w niektórych przypadkach nośna) na różnych elementach samolotu oprócz Skrzydeł. Na przykład wkład kadłuba do całkowitego oporu działającego na statek powietrzny. Dodatkowo, wszystko wystające z kadłuba przyczyni się do ogólnego oporu. Jeśli nie są to skrzydła, wszystko wystające z kadłuba jest zwykle nazywane dodatkiem. Niektóre przykłady przydatków to przyrządy do lądowania samolotów, baldachim, bomby, pociski, kapsuły paliwowe i wloty powietrza. Zwykle dane dotyczące przeciągania dla kadłubów i dodatków wyrażane są w postaci współczynnika oporu podobnego do tego omówionego w części 6, gdzie doświadczalnie określone siły oporu są niewymiarowane przez przewidywany obszar czołowy (S), gęstość (ρ) i prędkość do kwadratu (v^2). Oznacza to, że eksperymentalnie zmierzona siła oporu jest dzielona przez ilość $(1/2)\rho v^2 S$, aby uzyskać bezwymiarowy współczynnik oporu. W zależności od rozważanego obiektu c dane współczynnika oporu będą wyświetlane jako funkcja tego samego ważnego parametru geometrycznego, takiego jak kąt ataku w przypadku płatów lub stosunek długości do wysokości w przypadku daszków. Również tutaj Fluid Dynamie Drag firmy HQerner jest doskonałym źródłem praktycznych danych dla wszelkiego rodzaju kształtów i przydatków kadłuba. Na przykład, gdy podwozie statku powietrznego jest opuszczone, koła (a także powiązany z nimi mechanizm mechaniczny) przyczyniają się do ogólnej siły oporu na samolotach. Hoerner podaje współczynniki oporu w oparciu o powierzchnię czołową niektórych projektów małych podwozi płaskich w zakresie od 0,25 do 0,55. Dla porównania współczynniki oporu dla typowych zewnętrznych zbiorników magazynowych (na przykład dla paliwa), które zwykle są usprawnione, mogą wynosić od 0,06 do 0,26.

Kolejny składnik całkowitej siły oporu działającej na samoloty podczas lotu wynika z tarcia skóry. Skrzydła samolotu, kadłuby i dodatki nie są całkowicie gładkie. Spoina nity, a nawet farby, powodują

niedoskonałości powierzchni, które zwiększają opór tarcia. Podobnie jak w przypadku danych sfer przedstawionych w części 6, ten opór tarcia zależy od charakteru przepływu wokół rozpatrywanej części samolotu, to znaczy od tego, czy przepływ jest laminarny, czy turbulentny. Oznacza to, że współczynniki oporu tarcia dla konkretnych powierzchni będą na ogół zależne od liczby Reynoldsa. W rygorystycznej analizie konkretnego lotu samolotu, oczywiście chciałbyś wziąć pod uwagę wszystkie te dodatkowe elementy. Jeśli chciałbyś zobaczyć drobiazgowo szczegóły takich obliczeń, proponuję rzucić okiem na Fluid Dynamie Drag firmy Hoerner, w której podaje on dokładny przykład obliczenia całkowitej siły oporu dla myśliwca w jego części 14.

Kontrola

Kłapy znajdujące się na wewnętrznej krawędzi spływu skrzydła w naszym modelu służą do zmiany akordu i pochylenia sekcji skrzydła w celu zwiększenia siły nośnej przy danej prędkości. Kłapki służą przede wszystkim do zwiększenia siły nośnej podczas lotu w zwolnionym tempie, na przykład podczas startu lub lądowania. Podczas lądowania kłapy są zwykle rozstawiane pod dużym kątem do dołu (ugięcia kłap skierowane w dół są uważane za dodatnie) rzędu 30 do 60 stopni. Zwiększa to zarówno unoszenie się, jak i przeciąganie skrzydeł. Podczas lądowania ten wzrost oporu pomaga również w spowolnieniu samolotu do odpowiedniej prędkości lądowania. Podczas startu ten wzrost oporu działa przeciwko tobie, ponieważ wymaga większego ciągu, aby osiągnąć prędkość; w ten sposób kłapy nie mogą być rozmieszczane pod takim kątem, jak podczas lądowania. Lotki kontrolują lub wywołują ruch rolki poprzez produkcję różnicowego podniesienia między lotką a prawą stroną skrzydła. Podstawowa lotka jest niczym więcej jak parą krawędzi spływowych dopasowanych do końców skrzydeł. Kłapy te przesuwają się naprzeciwko siebie, jeden odchyła się w górę, podczas gdy drugi odchyła się w dół, aby utworzyć różnicę wzniosu między skrzydłami portu i prawej burty. Ta różnica wysokości, oddzielona odległością między lotkami, wytwarza moment obrotowy, który przetacza samolot. Aby przetoczyć samolot do. po lewej burcie pilota, lotka burtowa będzie odchylona w kierunku do dołu, podczas gdy lotka otworu zostanie odchylona w kierunku do góry względem pilota. Podobnie (przeciwnie ugięcia lotek spowodowałyby przewrócenie się na prawą burtę, w prawdziwym samolocie lotki są sterowane przez przesuwanie w lewo lub w prawo, windy, ogonowe "skrzydła" służą do sterowania nachyleniem samolotu, gdy windy są odchylone w taki sposób, że ich tylna krawędź skierowana jest w dół w stosunku do pilota, zostanie wywołany obrót nachylenia, to znaczy, że ogon samolotu będzie się podnosił w stosunku do jego nosa, a samolot będzie nurkował, w rzeczywistym samolocie osiąga się to poprzez popchnięcie drążka naprzód, a gdy windy zostaną odchylone tak, że ich krawędź ogonowa znajdzie się w górę, zostanie wywołany obrót nachylenia, a windy są bardzo ważne dla przycinania (Zasadniczo środek ciężkości samolotu znajduje się powyżej linii ćwierć akordowej skrzydeł samolotu, tak że środek ciężkości jest zgodny z główną siłą nośną. wcześniej, siła nośna nie zawsze przechodzi przez punkt ćwierćnutowy. Co więcej, środek ciężkości samolotu może bardzo dobrze się zmienić podczas lotu, na przykład, gdy paliwo jest spalane i kiedy uwalniana jest amunicja. Kontrolując windy, pilot jest w stanie dostosować ustawienie samolotu tak, że wszystkie siły równoważą się, a samolot leci w pożądanej orientacji (kącie nachylenia). Wreszcie, ster służy do kontroli odchylenia. Pilot używa pedałów do kontrolowania steru; naciskając lewy (port) pedał odchyła się w lewo, a pchnięcie prawych pedałów odchyła się w prawo (prawą burtę). Ster ten jest przydatny do precyzyjnego dostosowania ustawienia samolotu do podejścia do lądowania lub podczas obserwacji celu. Zazwyczaj duże działanie steru ma również wpływ na wychylenie rolki, które musi zostać skompensowane przez właściwe użycie lotek. W niektórych przypadkach ster składa się z kłapki na krawędzi spływu pionowego ogona; w innych przypadkach nie ma kłapy steru, a cały pionowy ogon obraca się. W obu przypadkach pionowy ogon, który zapewnia również kierunkową stabilność; zazwyczaj będzie miał symetryczny kształt płata; to znaczy, jego średnia linia krzywych będzie zbieżna z linią akordów. Kiedy samolot leci prosto i poziomo, ogon nie generuje wznoszenia, ponieważ jest symetryczny, a jego kąt natarcia wynosi zero.

Jeśli jednak płaszczyzna ślizga się (odchyła się w stosunku do kierunku lotu), wówczas ogon będzie pod kątem ataku i wygeneruje siłę nośną, zmierzając do pchnięcia płaszczyzny z powrotem do pierwotnej orientacji.

Modelowanie

Chociaż chcemy usunąć wiele materiałów! wymagane do wdrożenia symulatora lotu w czasie rzeczywistym, chciałbym przedstawić kilka kroków niezbędnych do obliczenia siły nośnej i siły uciążu na modelu samolotu. Oto kroki:

1. Dyskretyzuj powierzchnie podnoszące do kilku mniejszych sekcji skrzydeł.
2. Zbierz dane dotyczące wydajności geometrycznej płata.
3. Obliczyć względną zmienność powietrza nad każdą sekcją skrzydła.
4. Oblicz kąt natarcia dla każdej sekcji skrzydła.
5. Określ odpowiednie współczynniki siły nośnej i oporu oraz oblicz siły podnoszenia i oporu.

Pierwszy krok jest stosunkowo prosty, ponieważ musisz podzielić samolot na mniejsze sekcje, z których każda ma w przybliżeniu jednolity charakter. Wykonując ten krok dla modelu pokazanego na rysunku 2, można podzielić skrzydło na cztery sekcje: jedną dla każdej sekcji skrzydła, która jest wyposażona w lotkę i jedną dla każdej sekcji, która jest wyposażona w klapę. Można również użyć dwóch sekcji do modelowania steru wysokości - jeden port i jedna sterburta - i inna sekcja do modelowania ogona / steru. Wreszcie; można złożyć cały kadłub razem jako jedną dodatkową sekcję lub dalej podzielić go na mniejsze sekcje, w zależności od tego, jak szczegółowy ma być uzyskany. Jeśli zamierzasz modelować swój samolot jako sztywne ciało, będziesz musiał uwzględnić wszystkie siły i momenty działające na samolot podczas lotu. Ponieważ samolot składa się z wielu różnych elementów, z których każdy przyczynia się do całkowitej siły nośnej i oporu, będziesz musiał przerwać obliczenia na kilka mniejszych części, a następnie zsumować wszystkie składki, aby uzyskać wypadkowe siły unoszenia i przeciągania. Następnie możesz użyć tych wypadkowych sił wraz z pchnięciem i grawitacją w równaniach ruchu dla twojego samolotu. Możesz oczywiście jeszcze bardziej udoskonalić swój model, dodając więcej komponentów do takich elementów, jak czasza kokpitu, podwozie, zewnętrzne kapsuły paliwowe i bomby. Poziom szczegółowości zależy od stopnia dokładności, jaki zamierzasz osiągnąć. dla. Jeśli próbujesz naśladować osiągi konkretnego samolotu, musisz wyostrzyć ołówek. Po zdefiniowaniu każdej sekcji należy przygotować odpowiednie dane geometryczne i dane dotyczące wydajności. Na przykład w przypadku skrzydeł i innych powierzchni podnoszących należy określić początkowy kąt padania każdej sekcji (jej stały kąt nachylenia lub kąt natarcia w stosunku do referencyjnego systemu samolotu), rozpiętość, długość cięciwy, współczynnik kształtu, obszar wykresu i kwartał. umiejscowienie akordu względem środka ciężkości samolotu. Będziesz także musiał przygotować tabelę współczynników wzrostu i oporu w funkcji kąta natarcia odpowiedniego dla rozważanej sekcji. Ponieważ dane te są zwykle przedstawiane w formie graficznej, musisz pobrać dane z wykresów, aby zbudować tabelę odnośników do wykorzystania w twojej grze. Wreszcie; musisz obliczyć jednostkowy wektor normalny prostopadły do płaszczyzny każdej sekcji skrzydła. (Będziesz potrzebował tego później przy obliczaniu kąta natarcia.) Te pierwsze dwa kroki należy wykonać tylko raz na początku gry lub symulacji, ponieważ dane pozostaną niezmiennione (chyba że twój samolot zmieni kształt lub jego środek ciężkości przesunie się podczas symulacji). Trzeci krok polega na obliczeniu względnej prędkości między powietrzem i każdym komponentem, aby można było obliczyć siły podnoszenia i przeciągania; Na pierwszy rzut oka może wydawać się to banalne, ponieważ samolot będzie podróżował z prędkością powietrza, która będzie znana podczas symulacji. Należy jednak pamiętać, że statek powietrzny jest

sztynnym ciałem, a oprócz prędkości liniowej jego środka ciężkości należy również uwzględnić jego prędkość obrotową. W części drugiej przedstawiłem formułę do obliczenia prędkości względnej dowolnego punktu na sztywnym korpusie, który poruszał się zarówno liniowo, jak i obrotowo:

$$\mathbf{v}_R = \mathbf{v}_{cg} + (\boldsymbol{\omega} \times \mathbf{r})$$

To jest wzór, w którym musisz obliczyć prędkość względną w każdym komponencie w twoim modelu. W tym przypadku \mathbf{v}_{cg} jest wektorem reprezentującym prędkość lotu i kierunek lotu statku powietrznego, $\boldsymbol{\omega}$ jest wektorem prędkości kątowej statku powietrznego, a \mathbf{r} jest wektorem odległości od środka ciężkości statku powietrznego do rozważanego podzespołu. Kiedy mamy do czynienia ze skrzydłami, gdy mamy wektor prędkości względnej, można przystąpić do obliczenia kąta ataku dla każdej sekcji skrzydła. Wrzeczono siły oporu będzie równoległe do wektora prędkości względnej, podczas gdy wektor siły nośnej będzie prostopadły do wektora prędkości. Kąt natarcia jest wtedy kątem między wektorem siły nośnej a prostym wektorem prostopadłym do płaszczyzny skrzydła. Wymaga to pobrania iloczynu punktowego tych dwóch wektorów. Gdy masz już kąt ataku, możesz przejść do współczynnika wznoszenia i przeciągania względem ataku, tabel kątowych, aby określić współczynniki siły udźwigu i oporu do użycia w tej chwili w twojej symulacji. Za pomocą tych współczynników można użyć następujących wzorów do zobrazowania wielkości sił podnoszenia i przeciągania w rozpatrywanym odcinku skrzydła:

$$\text{Lift} = C_L(1/2)\rho V^2 S$$

$$\text{Drag} = C_D(1/2)\rho V^2 S$$

Przedstawione tutaj podejście jest bardzo uproszczonym podejściem, które jedynie przybliża charakterystykę podnoszenia i oporu. Podejście to nie uwzględnia efektów przepływu spontanicznego ani efektów przepływu między sąsiednimi sekcjami bocznymi. Takie podejście nie uwzględnia także zakłóceń powietrza, takich jak pranie, które może wpływać na względny kąt natarcia sekcji skrzydła. Ponadto przyjmuje się, że przepływ powietrza przez każdą sekcję skrzydła jest stały i jednorodny. Jako prosty przykład rozważ panel skrzydłowy 1, który jest prawym skrzydłem skrzydła lotki. Załóżmy, że skrzydło jest ustawione pod początkowym kątem nachylenia 3,5 stopnia i że samolot porusza się z prędkością 75 węzłów w locie poziomym na małej wysokości z kątem skoku wynoszącym 4,5 stopnia. Ten odcinek skrzydła ma długość cięciwy 5,2 stopy, a rozpiętość tego odcinka wynosi 6 stóp. Korzystając z danych dotyczących windy i oporu przedstawionych na rysunku 6, obliczyć siłę nośną i siłę ciągu w tej sekcji skrzydła, przyjmując, że lotki nie są odchylone, a gęstość powietrza $2,37 \times 10^{-3}$ slugs/ft³. Pierwszym krokiem jest obliczenie kąta natarcia, który wynosi 8 stopni, w oparciu o dostarczone informacje. Patrząc na rysunek 6, współczynniki unoszenia się i oporu można określić odpowiednio na 0,92 i 0,013. Następnie musisz obliczyć obszar planogramu w tej sekcji, który jest po prostu jego akord razy jego rozpiętości. To daje 31,2 ft². Teraz masz dość informacji, aby obliczyć siłę unoszenia i przeciągania jako towarzyszy (nie zapomnij przekształcić prędkości w węzłach na ft/s, 1 węzeł = 1,688 ft/s):

$$\text{Nośna} = C_L(1/2)\rho V^2 S$$

$$\text{Opór} = 0.92(1/2)(2.37 \times 10^{-3} \text{ slug/ft}^3)[(75 \text{ kt})(1.688 \text{ ft/s/kt})]^2(31.2 \text{ ft}^2)$$

$$\text{Nośna} = 542.2 \text{ lb}$$

$$\text{Opór} = C_D(1/2)\rho V^2 S$$

$$\text{Opór} = 0.013(1/2)(2.37 \times 10^{-3} \text{ slug/ft}^3)[(75 \text{ kt})(1.688 \text{ ft/s/kt})]^2(31.2 \text{ ft}^2)$$

$$\text{Opór} = 8.0 \text{ lb}$$

W twojej symulacji będziesz musiał wykonać podobny zestaw obliczeń dla każdego zdefiniowanego przez siebie komponentu. Jak widać, wykorzystanie tego rodzaju empirycznych danych i formuł, choć tylko przybliżone, pozwala na dość łatwe obliczenia. Trudną częścią jest ustalenie, które modele i znalezienia właściwych danych; po czacie windy i kalkulacje są dość proste. Przygotowałem przykładowy program pokazujący, jak modelować prosty samolot za pomocą pokazanej tutaj metody. Program nosi nazwę FlightSim .exe i jest trójwymiarowym symulatorem lotu w 3D .Mały symulator samolotu przypomina ten pokazany na rysunku 2. Ten program zawiera następujące pliki źródłowe wraz z plikiem te.txt (Instructions.txt), który wyjaśnia informacje o locie:

- Fizyka.cpp i Physi.cs.h
- DJdstuff.cpp i DJdstuff.h
- Mymath.h
- "Winmain.cpp

Jak już powiedziałem, program ten jest symulacją w czasie rzeczywistym, i traktuje samolot jako sztywne ciało. Nie uwzględniliśmy jeszcze symulacji w czasie rzeczywistym, więc wiele kodu może być mylących w tym momencie. Nie martw się jednak; później w tej książce omówię wszystko, co musisz wiedzieć, aby w pełni zrozumieć ten program. Na razie jednak chcę skupić się na kilku konkretnych funkcjach, które implementują model lotu. Funkcje te znajdują się w pliku źródłowym Physics.cpp.

Pierwszą funkcją jest CalcAirplaneMassProperties:

```

//-----//
// This model uses a set of eight discrete elements to represent the
// airplane. The elements are described below:
//
//      Element 1: Outboard, Port (left) wing section fitted with ailerons
//      Element 2: Inboard, Port wing section fitted with landing flaps
//      Element 3: Inboard, Starboard (right) wing section fitted with
//      landing flaps
//      Element 4: Outboard, Starboard wing section fitted with ailerons
//      Element 5: Port elevator fitted with flap
//      Element 6: Starboard elevator fitted with flap
//      Element 7: Vertical tail/rudder (no flap the whole thing rotates)
//      Element 8: The fuselage
//
// This function first sets up each element and then goes on to calculate
// the combined weight, center of gravity, and inertia tensor for the plane.
// Some other properties of each element are also calculated which you'll
// need when calculating the lift and drag forces on the plane.
//-----//
void CalcAirplaneMassProperties(void)
{
    float    mass;
    Vector    vMoment;
    Vector    CG;
    int       i;
    float     Ixx, Iyy, Izz, Ixy, Ixz, Iyz;
    float     in, di;

    // Initialize the elements here
    // Initially, the coordinates of each element are referenced from
    // a design coordinates system located at the very tail end of the plane,
    // its baseline, and its centerline. Later, these coordinates will be
    // adjusted so that each element is referenced to the combined center of
    // gravity of the airplane.
    Element[0].fMass = 6.56f;
    Element[0].vDCoords = Vector(14.5f,12.0f,2.5f);
    Element[0].vLocalInertia = Vector(13.92f,10.50f,24.00f);
    Element[0].fIncidence = -3.5f;
    Element[0].fDihedral = 0.0f;
    Element[0].fArea = 31.2f;
    Element[0].iFlap = 0;

    Element[1].fMass = 7.31f;
    Element[1].vDCoords = Vector(14.5f,5.5f,2.5f);
    Element[1].vLocalInertia = Vector(21.95f,12.22f,33.67f);
    Element[1].fIncidence = -3.5f;
    Element[1].fDihedral = 0.0f;
    Element[1].fArea = 36.4f;
    Element[1].iFlap = 0;

    Element[2].fMass = 7.31f;
    Element[2].vDCoords = Vector(14.5f,-5.5f,2.5f);
    Element[2].vLocalInertia = Vector(21.95f,12.22f,33.67f);
    Element[2].fIncidence = -3.5f;
    Element[2].fDihedral = 0.0f;
    Element[2].fArea = 36.4f;
    Element[2].iFlap = 0;
}

```

```

Element[3].fMass = 6.56f;
Element[3].vDCoords = Vector(14.5f,-12.0f,2.5f);
Element[3].vLocalInertia = Vector(13.92f,10.50f,24.00f);
Element[3].fIncidence = -3.5f;
Element[3].fDihedral = 0.0f;
Element[3].fArea = 31.2f;
Element[3].iFlap = 0;

Element[4].fMass = 2.62f;
Element[4].vDCoords = Vector(3.03f,2.5f,3.0f);
Element[4].vLocalInertia = Vector(0.837f,0.385f,1.206f);
Element[4].fIncidence = 0.0f;
Element[4].fDihedral = 0.0f;
Element[4].fArea = 10.8f;
Element[4].iFlap = 0;

Element[5].fMass = 2.62f;
Element[5].vDCoords = Vector(3.03f,-2.5f,3.0f);
Element[5].vLocalInertia = Vector(0.837f,0.385f,1.206f);
Element[5].fIncidence = 0.0f;
Element[5].fDihedral = 0.0f;
Element[5].fArea = 10.8f;
Element[5].iFlap = 0;

Element[6].fMass = 2.93f;
Element[6].vDCoords = Vector(2.25f,0.0f,5.0f);
Element[6].vLocalInertia = Vector(1.262f,1.942f,0.718f);
Element[6].fIncidence = 0.0f;
Element[6].fDihedral = 90.0f;
Element[6].fArea = 12.0f;
Element[6].iFlap = 0;

Element[7].fMass = 31.8f;
Element[7].vDCoords = Vector(15.25f,0.0f,1.5f);
Element[7].vLocalInertia = Vector(66.30f,861.9f,861.9f);
Element[7].fIncidence = 0.0f;
Element[7].fDihedral = 0.0f;
Element[7].fArea = 84.0f;
Element[7].iFlap = 0;

// Calculate the vector normal (perpendicular) to each lifting surface.
// This is required when calculating the relative air velocity for
// lift and drag calculations.
for (i = 0; i < 8; i++)
{
    in = DegreesToRadians(Element[i].fIncidence);
    di = DegreesToRadians(Element[i].fDihedral);
    Element[i].vNormal = Vector((float)sin(in), (float)(cos(in)*sin(di)),
                                (float)(cos(in)*cos(di)));
    Element[i].vNormal.Normalize();
}

// Calculate total mass
mass = 0;
for (i = 0; i < 8; i++)
    mass += Element[i].fMass;

// Calculate combined center of gravity location
vMoment = Vector(0.0f, 0.0f, 0.0f);
for (i = 0; i < 8; i++)

```

```

{
    vMoment += Element[i].fMass*Element[i].vDCoords;
}
CG = vMoment/mass;

// Calculate coordinates of each element with respect to the combined CG
for (i = 0; i < 8; i++)
{
    Element[i].vCGCoords = Element[i].vDCoords - CG;
}

// Now calculate the moments and products of inertia for the
// combined elements.
// (This inertia matrix (tensor) is in body coordinates)
Ixx = 0;    Iyy = 0;    Izz = 0;
Ixy = 0;    Ixz = 0;    Iyz = 0;
for (i = 0; i < 8; i++)
{
    Ixx += Element[i].vLocalInertia.x + Element[i].fMass *
        (Element[i].vCGCoords.y*Element[i].vCGCoords.y +
         Element[i].vCGCoords.z*Element[i].vCGCoords.z);
    Iyy += Element[i].vLocalInertia.y + Element[i].fMass *
        (Element[i].vCGCoords.z*Element[i].vCGCoords.z +
         Element[i].vCGCoords.x*Element[i].vCGCoords.x);
    Izz += Element[i].vLocalInertia.z + Element[i].fMass *
        (Element[i].vCGCoords.x*Element[i].vCGCoords.x +
         Element[i].vCGCoords.y*Element[i].vCGCoords.y);
    Ixy += Element[i].fMass * (Element[i].vCGCoords.x *
        Element[i].vCGCoords.y);
    Ixz += Element[i].fMass * (Element[i].vCGCoords.x *
        Element[i].vCGCoords.z);
    Iyz += Element[i].fMass * (Element[i].vCGCoords.y *
        Element[i].vCGCoords.z);
}

// Finally, set up the airplane's mass and its inertia matrix and take the
// inverse of the inertia matrix
Airplane.fMass = mass;
Airplane.mInertia.e11 = Ixx;
Airplane.mInertia.e12 = -Ixy;
Airplane.mInertia.e13 = -Ixz;
Airplane.mInertia.e21 = -Ixy;
Airplane.mInertia.e22 = Iyy;
Airplane.mInertia.e23 = -Iyz;
Airplane.mInertia.e31 = -Ixz;
Airplane.mInertia.e32 = -Iyz;
Airplane.mInertia.e33 = Izz;

Airplane.mInertiaInverse = Airplane.mInertia.Inverse();
}

```

Między innymi ta funkcja kończy się etapem 1 (i częścią kroku 2) naszej metody modelowania: dyskretyzuj samolot na wiele mniejszych elementów, z których każdy ma własną masę i właściwości lift i drag. Dla tego modelu zdecydowałem się użyć ośmiu elementów lub elementów, aby opisać samolot. Moje komentarze na początku funkcji wyjaśniają, co reprezentuje każdy element. Pierwszą rzeczą, jaką wykonuje ta funkcja, jest zainicjowanie elementów za pomocą właściwości, które zdefiniowałem w celu zbliżenia samolotu. Każdemu elementowi przypisuje się masę, zestaw współrzędnych projektowych do środka masy, zbiór momentów bezwładności dla środka ciężkości każdego elementu, początkowy kąt padania, obszar wykresu i kąt pomiaru. Współrzędne projektowe są współrzędnymi elementu względem punktu początkowego znajdującego się na samym wierzchołku samolotu tai !, na jego linii środkowej i na linii bazowej. Oś X tego układu wskazuje na nos samolotu, a oś y wskazuje na

stronę portu. Oś Z wskazuje w górę. Musisz najpierw skonfigurować swój układ współrzędnych, ponieważ nie znasz jeszcze położenia środka ciężkości całego samolotu, który jest połączeniem środka ciężkości wszystkich elementów. Ostatecznie chcesz, aby każdy element, do którego się odwołujesz, był połączony ze środkiem masy, ponieważ jest to środek masy, który będziesz śledzić podczas symulacji. (Przypomnijmy, że omówiliśmy to w części 2 i 4.) Kąt dwuścienny to kąt wokół osi x, na którym początkowo ustawiony jest element. Dla naszego modelu wszystkie elementy mają zerowy kąt dwuścienny, to znaczy są poziome, z wyjątkiem tai! ster, który ma 90-stopniowe dwuścienne, ponieważ jest zorientowany pionowo. Po ustawieniu elementów, pierwsze obliczenie, jakie wykonuje ta funkcja, polega na znalezieniu jednostki normalnej wektor do powierzchni każdego elementu na podstawie zapadalności elementu i kątów dwuściennych. Potrzebujesz wektora kierunku, aby obliczyć kąt natarcia między przepływem powietrza a elementem. Następnym obliczeniem jest obliczenie masy całkowitej, która jest po prostu sumą wszystkich mas elementów. Bezpośrednio po tym, połączone położenie środka ciężkości określa się za pomocą techniki, którą omówiłem w części 1. Współrzędne łączonego środka ciężkości odnoszą się do współrzędnych układu projektowego. Musisz odjąć tę współrzędną od współrzędnych projektowych każdego elementu, aby określić współrzędne każdego elementu względem łączonego środka ciężkości. Po tym wszystkim jesteś kompletny, z wyjątkiem łączonego momentu tensora bezwładności, który będę czekał, aż Part omówi. Krok 2 naszej metody modelowania mówi, że musisz zebrać dane dotyczące wydajności płata. Dla przykładu, do modelowania skrzydeł i wind użyliśmy profilowanego płata z gładkimi klapami, a do modelowania ogonowego steru użyłem symetrycznego płata bez klap. Nie użyłem klap do steru ogonowego, ponieważ właśnie obróciłem całość wokół pionowej osi, aby zapewnić działanie steru. Dla skrzydeł ustawiłem dwie funkcje do obsługi współczynników podnoszenia i oporu:

```
//-----//  
// Given the attack angle and the status of the flaps, this function  
// returns the appropriate lift coefficient for a cambered airfoil with  
// a plain trailing edge flap (+/- 15 degree deflection).  
//-----//  
float LiftCoefficient(float angle, int flaps)  
{  
    float cLfo[9] = {-0.54f, -0.2f, 0.2f, 0.57f, 0.92f, 1.21f, 1.43f, 1.4f,  
                    1.0f};  
}
```

```

float clfd[9] = {0.0f, 0.45f, 0.85f, 1.02f, 1.39f, 1.65f, 1.75f, 1.38f,
1.17f};
float clfu[9] = {-0.74f, -0.4f, 0.0f, 0.27f, 0.63f, 0.92f, 1.03f, 1.1f,
0.78f};
float a[9] = {-8.0f, -4.0f, 0.0f, 4.0f, 8.0f, 12.0f, 16.0f, 20.0f,
24.0f};
float cl;
int i;
cl = 0;
for (i=0; i<8; i++)
{
    if( (a[i] <= angle) && (a[i+1] > angle) )
    {
        switch(flaps)
        {
            case 0:// flaps not deflected
                cl = clfo[i] - (a[i] - angle) * (clfo[i] - clfo[i+1]) /
                (a[i] - a[i+1]);
                break;
            case -1: // flaps down
                cl = clfd[i] - (a[i] - angle) * (clfd[i] - clfd[i+1]) /
                (a[i] - a[i+1]);
                break;
            case 1: // flaps up
                cl = clfu[i] - (a[i] - angle) * (clfu[i] - clfu[i+1]) /
                (a[i] - a[i+1]);
                break;
        }
        break;
    }
}
return cl;
}

}

//-----//
// Given the attack angle and the status of the flaps, this function
// returns the appropriate drag coefficient for a cambered airfoil with
// a plain trailing edge flap (+/- 15 degree deflection).
//-----//
float DragCoefficient(float angle, int flaps)
{
    float cdf0[9] = {0.01f, 0.0074f, 0.004f, 0.009f, 0.013f, 0.023f, 0.05f,
0.12f, 0.21f};
    float cdfd[9] = {0.0065f, 0.0043f, 0.0055f, 0.0153f, 0.0221f, 0.0391f, 0.1f,
0.195f, 0.3f};
    float cdfu[9] = {0.005f, 0.0043f, 0.0055f, 0.02601f, 0.03757f, 0.06647f,
0.13f, 0.18f, 0.25f};
    float a[9] = {-8.0f, -4.0f, 0.0f, 4.0f, 8.0f, 12.0f, 16.0f, 20.0f,
24.0f};

    float cd;
    int i;

    cd = 0.5;
    for (i=0; i<8; i++)
    {
        if( (a[i] <= angle) && (a[i+1] > angle) )
        {
            switch(flaps)

```

```

    {
        case 0:// flaps not deflected
            cd = cdf0[i] - (a[i] - angle) * (cdf0[i] - cdf0[i+1]) /
                (a[i] - a[i+1]);
            break;
        case -1: // flaps down
            cd = cdfd[i] - (a[i] - angle) * (cdfd[i] - cdfd[i+1]) /
                (a[i] - a[i+1]);
            break;
        case 1: // flaps up
            cd = cdfu[i] - (a[i] - angle) * (cdfu[i] - cdfu[i+1]) /
                (a[i] - a[i+1]);
            break;
    }
    break;
}
}
return cd;
}

```

Każda z tych funkcji przyjmuje kąt ataku jako parametr wraz z flagą używaną do wskazania stanu klap, tj. Czy kłapy są w położeniu neuronalnym, odchylone w dół, czy odchylone w górę. Zauważ, że dane dotyczące siły nośnej i współczynnika oporu podane są dla zestawu dyskretnych kątów natarcia; w ten sposób interpolacja liniowa jest używana do określania współczynników dla kątów natarcia, które mieszczą się między dyskretnymi kątami. Funkcje określania współczynników wznoszenia i oporu steru ogonowego są podobne do tych pokazanych tutaj dla skrzydeł, z których jedynymi różnicami są same współczynniki oraz fakt, że ster na tail nie zawiera klap. Oto funkcje:

```

//-----//
// Given the attack angle this function returns the proper lift coefficient
// for a symmetric (no camber) airfoil without flaps.
//-----//
float RudderLiftCoefficient(float angle)
{
    float clfo[7] = {0.16f, 0.456f, 0.736f, 0.968f, 1.144f, 1.12f, 0.8f};
    float a[7] = {0.0f, 4.0f, 8.0f, 12.0f, 16.0f, 20.0f, 24.0f};
    float cl;
    int i;
    float aa = (float) fabs(angle);

    cl = 0;
    for (i=0; i<8; i++)
    {
        if( (a[i] <= aa) && (a[i+1] > aa) )
        {
            cl = clfo[i] - (a[i] - aa) * (clfo[i] - clfo[i+1]) /
                (a[i] - a[i+1]);
            if (angle < 0) cl = -cl;
            break;
        }
    }
    return cl;
}

```



```

//-----//
// Given the attack angle this function returns the proper drag coefficient
// for a symmetric (no camber) airfoil without flaps.
//-----//
float RudderDragCoefficient(float angle)
{
    float cdf0[7] = {0.0032f, 0.0072f, 0.0104f, 0.0184f, 0.04f, 0.096f, 0.168f};
    float a[7] = {0.0f, 4.0f, 8.0f, 12.0f, 16.0f, 20.0f, 24.0f};
    float cd;
    int i;
    float aa = (float) fabs(angle);

    cd = 0.5;
    for (i=0; i<8; i++)
    {
        if( (a[i] <= aa) && (a[i+1] > aa) )
        {
            cd = cdf0[i] - (a[i] - aa) * (cdf0[i] - cdf0[i+1]) /
                (a[i] - a[i+1]);
            break;
        }
    }
    return cd;
}

```

W przypadku kroków 1 i 2 na drodze, kroki 3, 4 i 5 są obsługiwane w jednej funkcji obciążonej obciążeniami CalcAirplane:

```

//-----//
// This function calculates all of the forces and moments acting on the
// plane at any given time.
//-----//
void CalcAirplaneLoads(void)
{
    Vector Fb, Mb;

    // reset forces and moments:
    Airplane.vForces.x = 0.0f;
    Airplane.vForces.y = 0.0f;
    Airplane.vForces.z = 0.0f;

    Airplane.vMoments.x = 0.0f;
    Airplane.vMoments.y = 0.0f;
    Airplane.vMoments.z = 0.0f;

    Fb.x = 0.0f; Mb.x = 0.0f;
    Fb.y = 0.0f; Mb.y = 0.0f;
    Fb.z = 0.0f; Mb.z = 0.0f;

    // Define the thrust vector, which acts through the plane's CG
    Thrust.x = 1.0f;
    Thrust.y = 0.0f;
    Thrust.z = 0.0f;
    Thrust *= ThrustForce;

    // Calculate forces and moments in body space:
    Vector vLocalVelocity;
    float fLocalSpeed;
    Vector vDragVector;
    Vector vLiftVector;
    float fAttackAngle;
}

```

```

float    tmp;
Vector   vResultant;
int      i;
Vector   vtmp;

Stalling = false;

for(i=0; i<7; i++) // loop through the seven lifting elements
    // skipping the fuselage
{
    if (i == 6) // The tail/rudder is a special case, since it can rotate;
    {
        // therefore, you have to recalculate the normal vector
        float in, di;
        in = DegreesToRadians(Element[i].fIncidence); // incidence angle
        di = DegreesToRadians(Element[i].fDihedral); // dihedral angle
        Element[i].vNormal = Vector(
            (float)sin(in),
            (float)(cos(in)*sin(di)),
            (float)(cos(in)*cos(di)));

        Element[i].vNormal.Normalize();
    }

    // Calculate local velocity at element
    // The local velocity includes the velocity due to linear
    // motion of the airplane,
    // plus the velocity at each element due to the
    // rotation of the airplane.

    // Here's the rotational part
    vtmp = Airplane.vAngularVelocity^Element[i].vCGCoords;

    vLocalVelocity = Airplane.vVelocityBody + vtmp;

    // Calculate local air speed
    fLocalSpeed = vLocalVelocity.Magnitude();

    // Find the direction in which drag will act.
    // Drag always acts inline with the relative
    // velocity but in the opposing direction
    if(fLocalSpeed > 1.)
        vDragVector = -vLocalVelocity/fLocalSpeed;

    // Find the direction in which lift will act.
    // Lift is always perpendicular to the drag vector
    vLiftVector = (vDragVector^Element[i].vNormal)^vDragVector;
    tmp = vLiftVector.Magnitude();
    vLiftVector.Normalize();

    // Find the angle of attack.
    // The attack angle is the angle between the lift vector and the
    // element normal vector. Note that the sine of the attack angle,
    // is equal to the cosine of the angle between the drag vector and
    // the normal vector.
    tmp = vDragVector*Element[i].vNormal;
    if(tmp > 1.) tmp = 1;
    if(tmp < -1) tmp = -1;
    fAttackAngle = RadiansToDegrees((float) asin(tmp));

    // Determine the resultant force (lift and drag) on the element.
    tmp = 0.5f * rho * fLocalSpeed*fLocalSpeed * Element[i].fArea;
    if (i == 6) // Tail/rudder

```

```

    {
        vResultant = (vLiftVector*RudderLiftCoefficient(fAttackAngle) +
                    vDragVector*RudderDragCoefficient(fAttackAngle))
                    * tmp;
    } else
        vResultant = (vLiftVector*LiftCoefficient(fAttackAngle,
            Element[i].iFlap) +
                    vDragVector*DragCoefficient(fAttackAngle,
            Element[i].iFlap) ) * tmp;
    // Check for stall.
    // We can easily determine when stalled by noting when the coefficient
    // of lift is zero. In reality stall warning devices give warnings well
    // before the lift goes to zero to give the pilot time to correct.
    if (i<=0)
    {
        if (LiftCoefficient(fAttackAngle, Element[i].iFlap) == 0)
            Stalling = true;
    }

    // Keep a running total of these resultant forces (total force)
    Fb += vResultant;

    // Calculate the moment about the CG of this element's force
    // and keep a running total of these moments (total moment)
    vtmp = Element[i].vCGCoords*vResultant;
    Mb += vtmp;
}

// Now add the thrust
Fb += Thrust;

// Convert forces from model space to earth space
Airplane.vForces = QVRotate(Airplane.qOrientation, Fb);

// Apply gravity (g is defined as -32.174 ft/s 2)
Airplane.vForces.z += g * Airplane.fMass;

Airplane.vMoments += Mb;
}

```

Pierwszą rzeczą, jaką wykonuje ta funkcja, jest reset zmiennych, które posiadają całkowitą siłę i moment działające na samolot. Następnie ustawia się wektor ciągu. W tym przykładzie jest to trywialne, ponieważ zakładam, że wektor ciągu zawsze wskazuje w kierunku dodatnim w kierunku osi x (w kierunku nosa) i przechodzi przez środek ciężkości samolotu (więc nie tworzy chwili). Po obliczeniu wektora ciągu funkcja wykonuje pętlę nad elementami modelu, aby obliczyć siłę unoszenia i oporu dla każdego elementu. W tym modelu pominąłem kadłub; Jeśli jednak chcesz uwzględnić jego opór w modelu, jest to miejsce, w którym należy dodać obliczanie oporu. Wchodząc w pętlę, pierwszą rzeczą, jaką wykonuje funkcja, jest sprawdzenie, czy bieżącym elementem jest element nr 6, ogon steru. Jeśli tak, to norma steru! wektor jest przeliczany na podstawie aktualnego kąta padania. Kąt padania steru zmienia się po naciśnięciu klawisza X lub C, aby zastosować działanie steru kierunku. Następne obliczenie polega na wyznaczeniu względnej prędkości między powietrzem a rozpatrywanym elementem. Jak już wcześniej wspominałem, ta prędkość względna składa się z prędkości liniowej, gdy samolot porusza się w powietrzu plus prędkość dowolnego elementu z powodu obrotu samolotu. Po uzyskaniu tego wektora obliczysz relatywną prędkość powietrza, przyjmując wartość wektora prędkości względnej. Następnym krokiem jest określenie kierunku działania przecięgania. Ponieważ opór przeciwstawia się ruchowi, działa on zgodnie z przeciwną do wektora prędkości względnej; tak więc wszystko, co musisz zrobić, to wziąć ujemny wektor prędkości względnej i znormalizować wynik (podzielić go przez jego wielkość), aby uzyskać wektor kierunku przecięgania. Ponieważ wektor ten został znormalizowany, jego długość jest równa 1 (jedność), więc można go pomnożyć przez siłę oporu,

która zostanie obliczona później, aby uzyskać wektor siły przeciągania. Po uzyskaniu wektora kierunku przeciągania funkcja ta wykorzystuje go do określenia wektora kierunku podnoszenia. Wektor siły nośnej jest zawsze w kierunku prostopadłym do wektora siły oporu, więc aby obliczyć jego kierunek, najpierw przecinamy produkt krzyżyka wektora kierunku przeciągania z normalnym wektorem elementu, a następnie ponownie przekreślamy wynik za pomocą wektora kierunku przeciągania. Ponownie funkcja normalizuje wektor kierunku nośności. Teraz, gdy uzyskano wektory kierunku wznoszenia i przeciągania, funkcja oblicza kąt natarcia dla bieżącego elementu. Kąt ataku to kąt pomiędzy wektorem wzniosu a elementem normalnym. Kąt można obliczyć, biorąc \cdot cosinus odwrotny wektora iloczynu wektorowego kierunku wznoszenia za pomocą wektora normalnego elementu. Ponieważ wektor przeciągania jest prostopadły do wektora nośnika, można uzyskać ten sam wynik, biorąc sinus odwrotny wektora iloczynu wektorowego kierunku przeciągania z wektorem normalnym elementu. Teraz, gdy wszystkie przeszkody i elementy wektorowe przeciągają się z drogi, funkcja oblicza wypadkową siłę działającą na element. Wynikowy wektor siły jest po prostu sumą wektorów siły udźwigu i oporu. Zauważ, że jest to miejsce, w którym wywoływane są funkcje dźwigu i współczynnika oporu oraz gdzie stosowane są empiryczne formuły dźwigu i oporu, omówione wcześniej. Po obliczeniu siły wyniku, funkcja sprawdza, czy obliczony współczynnik wzrostu wynosi zero. Jeśli współczynnik wznoszenia wynosi zero, wówczas flaga przeciągnięcia jest ustawiona, aby ostrzegać nas, że samolot znajduje się w stanie zatrzymania. Ostatecznie, siła wypadkowa jest akumulowana w całkowitej zmiennej wektora siły, a moment jest obliczany przez pobranie iloczynu wektora współrzędnych wektora elementu z siłą wypadkową. Wynikowy moment jest kumulowany w całkowitej zmiennej wektorowej momentu. Po wyjściu z pętli funkcja dodaje wektor ciągu do całkowitej siły. Jak dotąd wszystkie te siły i momenty zostały odniesione w ustalonym układzie współrzędnych ciała. Jedyne, co pozostało do zrobienia, to zastosować siłę grawitacji, ale ta siła działa w ujemnym kierunku osi y w ułożonym układzie współrzędnych Ziemi. Aby zastosować siłę grawitacji, musi najpierw obrócić wektor siły ciała z przestrzeni ciała do współrzędnych przestrzeni Ziemi. W tym przykładzie użyłem techniki obrotu kwaternionowego, którą omówię później. To prawie wszystko w przypadku modelu lotu. Reszta kodu w tym przykładzie zostanie omówiona w dalszej części tekstu, tam gdzie jest to stosowne. Jediną rzeczą, której nie omówię, jest kod implementujący aspekty Direct3D tego programu. Śmiało i dostosuj właściwości elementu i zobacz, co się stanie. Mimo że jest to trudny model, wyniki lotu wyglądają całkiem realistycznie.