

---

# Fizyka dla grafików komputerowych



**KAPITAŁ LUDZKI**  
NARODOWA STRATEGIA SPÓJNOŚCI



**UMCS**  
UNIWERSYTET MEDYCYNICZNY  
W LUBLINIE

**UNIA EUROPEJSKA**  
EUROPEJSKI  
FUNDUSZ SPOŁECZNY



Projekt „Programowa i strukturalna reforma systemu kształcenia na Wydziale Mat-Fiz-Inf”.  
Projekt współfinansowany ze środków Unii Europejskiej w ramach Europejskiego Funduszu Społecznego.

Człowiek-najlepsza inwestycja



UNIwersYTET MARIi CURIE-SKŁODOWSKIEJ  
WYDZIAŁ MATEMATYKI, FIZYKI I INFORMATYKI  
INSTYTUT INFORMATYKI

# Fizyka dla grafików komputerowych

Paweł Mikołajczak



LUBLIN 2012

**Instytut Informatyki UMCS**

**Lublin 2012**

Paweł Mikołajczak

FIZYKA DLA GRAFIKÓW KOMPUTEROWYCH

**Recenzent:** Michał Chlebiej

Opracowanie techniczne: Marcin Denkowski

Projekt okładki: Agnieszka Kuśmierska

Praca współfinansowana ze środków Unii Europejskiej w ramach  
Europejskiego Funduszu Społecznego

Publikacja bezpłatna dostępna on-line na stronach  
Instytutu Informatyki UMCS: [informatyka.umcs.lublin.pl](http://informatyka.umcs.lublin.pl)

**Wydawca**

Uniwersytet Marii Curie-Skłodowskiej w Lublinie

Instytut Informatyki

pl. Marii Curie-Skłodowskiej 1, 20-031 Lublin

Redaktor serii: prof. dr hab. Paweł Mikołajczak

www: [informatyka.umcs.lublin.pl](http://informatyka.umcs.lublin.pl)

email: [dyrii@hektor.umcs.lublin.pl](mailto:dyrii@hektor.umcs.lublin.pl)

**Druk**

FIGARO Group Sp. z o.o. z siedziba w Rykach

ul. Warszawska 10

08-500 Ryki

www: [www.figaro.pl](http://www.figaro.pl)

ISBN: 978-83-62773-27-5

# SPIS TREŚCI

<b>PRZEDMOWA.....</b>	<b>VII</b>
<b>1. PODSTAWOWE KONCEPCJE FIZYKI .....</b>	<b>1</b>
1.1. Pomiary i jednostki .....	2
1.2. Układ współrzędnych i wektory .....	3
1.3. Siła i zasady dynamiki Newtona .....	15
1.4. Punkt materialny, środek masy, moment bezwładności.....	24
<b>2. RACHUNEK RÓŻNICZKOWY I CAŁKOWY .....</b>	<b>35</b>
2.1. Modele algebraiczne zjawisk fizycznych.....	36
2.2. Pochodne i różniczkowanie .....	38
2.3. Rachunek całkowy .....	42
2.4. Rachunek różnicowy .....	47
<b>3. METODY NUMERYCZNE .....</b>	<b>49</b>
3.1. Szereg Taylora .....	50
3.2. Proste metody całkowania numerycznego .....	51
3.3. Rozwiązywanie równań różniczkowych zwyczajnych metodami numerycznymi.....	62
<b>4. RUCH OBIEKTÓW W POLU GRAWITACYJNYM .....</b>	<b>75</b>
4.1. Równania ruchu .....	76
4.2. Równanie ruchu ze stałym przyspieszeniem.....	78
4.3. Rzut pionowy .....	79
4.4. Rzut poziomy i ukośny.....	83
<b>5. RUCH OBIEKTÓW Z UWZGLĘDNIENIEM OPORÓW POWIETRZA .....</b>	<b>89</b>
5.1. Siła oporu .....	90
5.2. Spadek pionowy z uwzględnieniem oporu powietrza .....	94
5.3. Rzut poziomy z uwzględnieniem oporu powietrza .....	98
5.4. Rzut ukośny z uwzględnieniem oporu powietrza.....	101
5.5. Efekt Magnusa i ruch obiektów .....	109
<b>6. MODELOWANIE RUCHU POJAZDÓW .....</b>	<b>115</b>
6.1. Czynniki wpływające na ruch pojazdów .....	116
6.2. Kinetyka w opisie ruchu samochodów.....	117

6.3. Siły tarcia w ruchu samochodów .....	122
6.4. Siła dośrodkowa w ruchu samochodów .....	125
<b>7. MODELOWANIE ZDERZEŃ .....</b>	<b>137</b>
7.1. Zderzenia obiektów i wykrywanie kolizji. ....	138
7.2. Elementy fizyki zderzeń. ....	139
7.3. Metody wykrywania kolizji. ....	152
<b>BIBLIOGRAFIA.....</b>	<b>159</b>

# PRZEDMOWA

Tworzenie realistycznych gier komputerowych wymaga stosowania zasad fizyki. Jeszcze do niedawna, ten postulat nie mógł być realizowany z powodu niewystarczających mocy obliczeniowych domowych komputerów. Obecny postęp w technice komputerowej (szybkie procesory, duże pamięci, wyspecjalizowane kary graficzne z GPU, nowoczesne technologie typu CUDA) pozwalają w coraz szerszym stopniu wykorzystywać formalizm fizyki do realistycznego modelowania zjawisk. Tworząc realistyczne gry komputerowe mamy zbiór wymagań:

- Chcemy, aby w naszej grafice, obiekty poruszały się w sposób naturalny
- Żądamy by ruch był zgodny z naszym codziennym doświadczeniem
- Musimy brać pod uwagę uwarunkowania (ograniczenia) sprzętowe
- Przetwarzanie nie może być zbyt kosztowne
- Musimy wyznaczyć granice dokładności modelowania i symulacji (korzystając z rachunku różniczkowego)

Fizyka jest rozległą i skomplikowaną dziedziną wiedzy, operującą skomplikowanym aparatem matematycznym. Na szczęście w praktycznym wykorzystaniu zasad fizyki w grach komputerowych i animacjach wystarczą wiadomości wykładane w szkole średniej. Kluczowym działem fizyki wykorzystywanym w grach komputerowych jest mechanika.

W uniwersyteckich programach nauczania grafika komputerowa jest istotnym elementem wykształcenia informatyka.

Niniejszy podręcznik przeznaczony jest dla studentów informatyki specjalizujących się w grafice komputerowej i tworzeniu gier komputerowych.

Podręcznik ma służyć pomocą studentom tworzącym oprogramowanie na zajęciach laboratoryjnych z zakresu gier i animacji komputerowych.

Od czytelnika niniejszego podręcznika wymaga się znajomości wybranego środowiska programistycznego (C++/ QT, Java).

Oczywiście znajomość algebry, analizy matematycznej i geometrii analitycznej jest także wymagana.

Podręcznik składa się z siedmiu części.

W rozdziale pierwszym omówiono podstawowe koncepcje fizyki: pomiary, jednostki wielkości fizycznych, pojęcie siły i zasad dynamiki Newtona oraz wprowadzono pojęcie punktu materialnego.

W rozdziale drugim omówiono niezbędny w animacjach i tworzeniu realistycznych gier komputerowych formalizm matematyczny: rachunek różniczkowy i całkowy.

Rozdział trzeci poświęcony jest metodom numerycznym, dużo uwagi poświęcono całkowaniu numerycznemu i numerycznym metodom rozwiązywania równań różniczkowych zwyczajnych.

W rozdziale czwartym omówiono ruch obiektów w ziemskim polu grawitacyjnym, opisano modelowanie rzutu pionowego, poziomego i ukośnego.

W rozdziale piątym ponownie omawiano rzuty, tym razem z uwzględnieniem wpływu oporu na ruch ciał.

Rozdział szósty omawia modelowanie ruchu pojazdów z uwzględnieniem wpływu tarcia, oporu i sił dośrodkowych.

W rozdziale siódmym omówiono modele zderzeń oraz metody wykrywania kolizji.

Podręcznik powstał na podstawie dziesięcioletniego doświadczenia autora w prowadzeniu wykładów i ćwiczeń laboratoryjnych z przedmiotu grafika komputerowa oraz wykładów z języków programowania i metod numerycznych prowadzonych na kierunku Informatyka w Uniwersytecie M. Curie-Skłodowskiej w Lublinie.



---

# ROZDZIAŁ 1

## PODSTAWOWE KONCEPCJE FIZYKI

---

1.1. Pomiary i jednostki .....	2
1.2. Układ współrzędnych i wektory.....	3
1.3. Siła i zasady dynamiki Newtona .....	15
1.4. Punkt materialny, środek masy, moment bezwładności.....	24

---

## 1.1. Pomiary i jednostki

Fizyka jest nauką o budowie oraz właściwościach materii i działających na nią siłach. Dzięki obserwacjom i doświadczeniom odkrywamy prawa rządzące w naturze. Zasadniczo fizyka opiera się na pomiarach. W doświadczeniach mierzymy różne wielkości. Każdą wielkość mierzymy w odpowiednich jednostkach. Jednostka to nazwa miary danej wielkości. Na przykład jednostką czasu jest sekunda, jednostką długości jest metr. Należy pamiętać, że mamy dość duże zamieszanie, jeżeli chodzi o stosowane jednostki pomiarowe. Na przykład w Stanach Zjednoczonych do dzisiaj temperaturę podaje się w stopniach Fahrenheita, a w Europie w stopniach Celsjusza. W 1792 roku we Francji wprowadzono nowy układ miar i wag – układ metryczny. Wtedy zdefiniowano metr, początkowo, jako jedną dziesięciomilionową część odległości od bieguna północnego do równika. Ze względów praktycznych zbudowano wzorzec metra - był to pręt wykonany ze stopu platyny i irydu, a metr określono, jako odległość między dwoma wygrawerowanymi kreskami na tym pręcie. W 1971 roku stworzono Międzynarodowy Układ Jednostek, nazywany układem SI. Ten system obowiązuje aktualnie. Wprowadzono także szereg zmian w definicjach konkretnych jednostek pomiarowych. Na przykład wzorzec metra zdefiniowany jest, jako 1650763,73 długości fali pomarańczowoczerwonej linii, wysyłanej przez atomy kryptonu 86 podczas wyładowania w tym gazie. Ale to nie koniec fascynującej historii metra. W 1983 roku ustalono, że metr jest to długość drogi, którą przebywa światło w próżni w czasie  $1/299792458$  sekundy. I ta definicja obowiązuje dzisiaj (rok 2012). W układzie SI możemy wyróżnić tzw. jednostki podstawowe. Są to metr (jednostką jest metr, symbol L), czas (jednostką jest sekunda, symbol T) i masa (jednostką jest kilogram, symbol M). Za pomocą jednostek podstawowych zdefiniowane są jednostki pochodne. W tabeli 1.1 podano wybrane jednostki układu SI.

Tabela 1.1. Wybrane jednostki układu SI

Wielkość fizyczna	symbol	wymiar	Jednostka SI
Długość	L	L	metr, m
Czas	T	T	sekunda, s
Masa	M	M	kilogram, kg
Siła	F	$M(L/T^2)$	niuton, N
Moment siły	M	$M(L^2/T^2)$	N*m
Prędkość liniowa	V	L/T	m/s
Prędkość kątowna	$\omega$	radian/T	radian/s
Przyspieszenie liniowe	a	$L/T^2$	$m/s^2$
Przyspieszenie kątowne	$\alpha$	radian/T <sup>2</sup>	radian/s <sup>2</sup>
Gęstość	$\rho$	$M/L^3$	$kg/m^3$
Moment bezwładności	I	$ML^2$	$kg*m^2$

W praktycznych zastosowaniach, bardzo często mamy do czynienia z zagadnieniem zamiany jednostek, w których wyrażona jest jakaś wielkość fizyczna.

Przykład 1.

Zmierzone czas, który wnosi 320 sekund. Ile to będzie minut?

Wiemy, że 1 minuta to 60 sekund. Wobec tego:

$$320s = 320s \cdot \frac{1min}{60s} = \frac{320}{60}min = 5.3 min$$

Przykład 2.

Gęstość wody w temperaturze pokojowej wynosi  $1.0g/cm^3$ . Ile wynosi gęstość wody w jednostkach  $kg/m^3$  ?

Wiemy, że  $1 kg = 1000 g$  oraz  $1m = 100 cm$ . Wobec tego mamy:

$$\frac{1g}{cm^3} = \frac{1g}{cm^3} \cdot \frac{1kg}{10^3g} \cdot \frac{(10^2cm)^3}{m^3} = 1.0 \cdot 10^3 \frac{kg}{m^3}$$

Przykład 3.

Samochód jedzie z prędkością 60 mil na godzinę (mi/g). Jaka jest ta prędkość w jardach na sekundę i metrach na sekundę?

Wiemy, że 1 mila to 1760 jardów (yd), a godzina ma 3600 sekund. Wobec tego mamy:

$$\frac{60 mi}{g} = \frac{60mi}{g} \cdot \frac{1760yd}{mi} \cdot \frac{1g}{3600s} = 29 \frac{yd}{s}$$

Wiemy także, że 1 jard to 0.9144 metrów, wobec tego mamy:

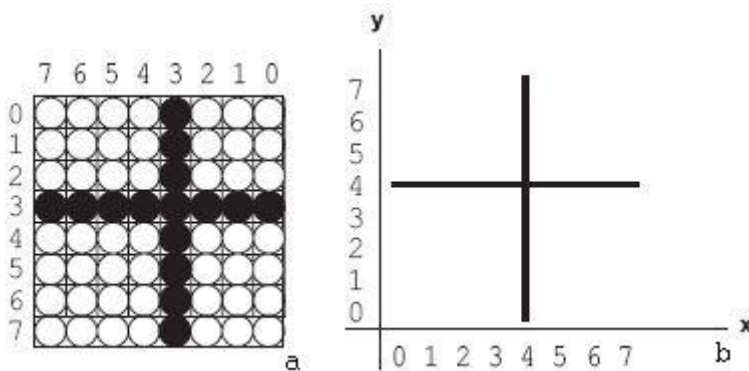
$$29 \frac{yd}{s} = 29 \frac{yd}{s} \cdot \frac{0.9144m}{yd} = 27 \frac{m}{s}$$

Stosowanie poprawnych jednostek ma zasadnicze znaczenie w modelowaniu. Zdarza się, że we wzorach użyte zostaną wielkości wyrażone w różnych jednostkach, wtedy wynik będzie całkowicie błędny. Zawsze należy sprawdzić stosowane wzory i sprawdzać zgodność jednostek.

## 1.2. Układ współrzędnych i wektory

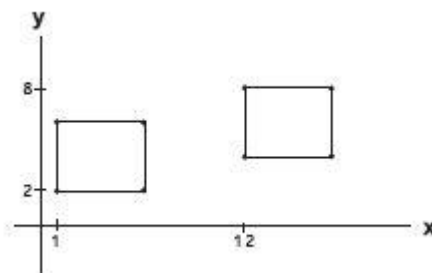
Fundamentalnym pojęciem w fizyce jest przestrzeń. W trójwymiarowej przestrzeni umieszczone są wszystkie materialne obiekty. Możemy opisać położenie obiektu, jeżeli zdefiniujemy jakiś system. Jak dotąd, takim eleganckim system jest kartezjański układ współrzędnych. W grafice komputerowej najczęściej operujemy dwuwymiarowym układem współrzędnych.

Mamy dwa typy obrazów komputerowych: definiowane jako mapy pikselowe (bitmapy) oraz definiowane analitycznie. W pierwszym przypadku mówimy o grafice rastrowej, w drugim mówimy o grafice wektorowej. Na rysunku 1.2.1 mamy przedstawione dwa obrazy (przecinające się pod kątem prostym dwa odcinki), obraz (a) zdefiniowany jest, jako bitmapa, obraz (b) wykonany jest techniką wektorową. Płaszczyzna rastrowa zbudowana jest z elementów zwanych pikselami. Tworzą one macierz wierszy i kolumn, dzięki czemu można kodować położenie każdego piksela. Można kodować obraz (na przykład w reprezentacji bitowej (0 – czarny piksel, 1 – biały piksel)). W ten sposób kodując wybrane piksele tworzymy obraz rastrowy dwóch przecinających się odcinków. Na obrazie (b) przecinające się linie zostały narysowane na podstawie poleceń wykreślenia linii. Linie wykreślamy podając ich współrzędne początku i końca w kartezjańskim układzie współrzędnych związanym z systemem wyświetlającym obrazy (np. monitor komputerowy).



Rys. 1.2.1 Przykłady grafiki rastrowej (a) i wektorowej (b)

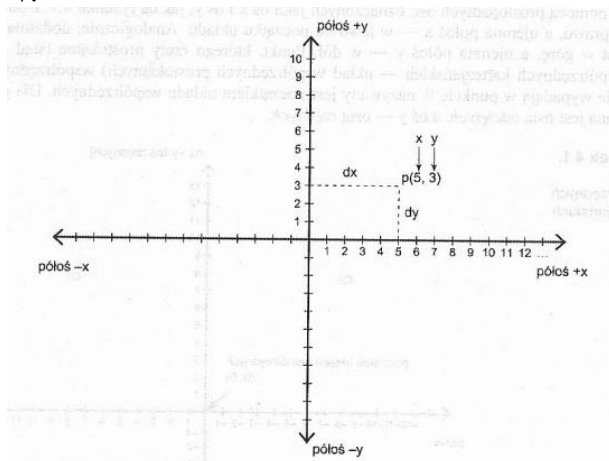
Każda metoda (rastrowa lub wektorowa) ma swoje wady i zalety. W grach komputerowych, ze względu na dużą ilość rysowanych detali, najczęściej wykorzystywana jest grafika rastrowa.



Rys. 1.2.2 Przesunięcie obiektu realizowane jest przez zmianę współrzędnych

Rendering 3D w całości jest oparty na transformacjach obiektów modelowanych punktami zdefiniowanymi we współrzędnych kartezjańskich. Na przykład przesunięcie obiektu na ekranie realizowane jest przez dodanie stałej do każdej współrzędnej punktu obiektu. Ta technika zilustrowana jest na rys. 1.2.2.

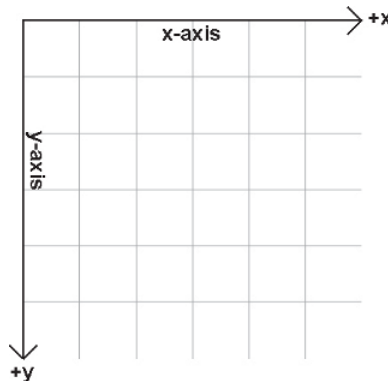
Koncepcja układu współrzędnych znakomicie ułatwia posługiwanie się pojęciami geometrii i fizyki. Układ współrzędnych dwuwymiarowych upowszechnił francuski matematyk Rene Descartes (1596 – 1650), znany w polskich podręcznikach, jako Kartezjusz. W swoim dziele zatytułowanym *La Geometrie*, wydanym w 1637 roku, Kartezjusz zastosował metodę analityczną. W metodzie analitycznej, posługujemy się pojęciem punktu, który wyznaczony jest przy pomocy pary liczb, zwanych współrzędnymi oraz na opisanie obiektu geometrycznego równaniem (lub równaniami). Na przykład dla okręgu można napisać równanie, które jest spełnione przez współrzędne punktu, gdy punkt należy do okręgu. W dwuwymiarowym układzie współrzędnych obieramy na płaszczyźnie dwie wzajemnie prostopadłe osie liczbowe. Pozioma oś oznaczona jako  $x$  nazywana jest osią odciętych (ang. *abscissa*), pionowa oś oznaczana jako  $y$  nazywana jest osią rzędnych (ang. *ordinate*). Dowolny punkt  $P$  można rzutować na osie układu współrzędnych. Położenie rzutów na osiach oznaczanych jako  $(x,y)$  nazywamy parą współrzędnych punktu  $P$  na płaszczyźnie  $Oxy$ . Liczbę  $x$  nazywamy odciętą punktu  $P$ , liczbę  $y$  nazywamy rzędną punktu  $P$ . Na rysunku 1.2.3 przedstawiono układ współrzędnych z zaznaczonym punktem  $P(2,7)$ . Przykładowy punkt  $P$  ma współrzędną  $x = 2$ , współrzędną  $y = 7$ .



Rys. 1.2.3 Lokalizacja punktu  $P$  w układzie współrzędnych kartezjańskich

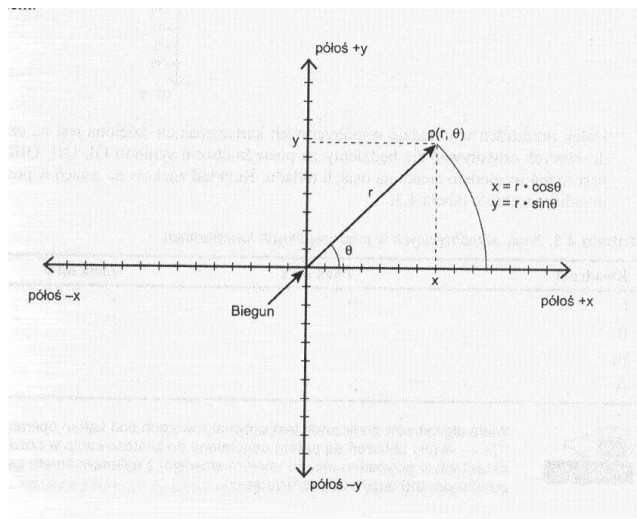
Wybór początku układu współrzędnych jest arbitralny. Zazwyczaj w grafice komputerowej układ współrzędnych związanych z ekranem monitora komputerowego ma postać pokazaną na rys.1.2.4. Należy pamiętać, że początek układu współrzędnych w tym systemie umieszczony jest w górnym lewym rogu ekranu.

Kierunek dodatni osi x skierowany jest w prawą stronę, a kierunek dodatni osi y skierowany jest z góry na dół.



Rys. 1.2.4 Układ współrzędnych kartezjańskich związanych z monitorem

Dość często w grafice komputerowej wykorzystywany jest inny układ współrzędnych – układ współrzędnych biegunowych (rys. 1.2.5). Ten układ był wykorzystany w grze komputerowej Wolfenstein do rzutowania promieni. Lokalizacja punktu P w układzie współrzędnych biegunowych polega na podaniu odległości punktu  $r$  od początku układu współrzędnych (bieguna) oraz wyznaczeniu kąta  $\theta$  jaki tworzy prosta OP z osią x. Należy pamiętać, że wartość kąta podajemy najczęściej w radianach, a sam kąt mierzony jest w kierunku przeciwnym do kierunku ruchu wskazówek zegara. Konwersje współrzędnych punktu z jednego układu współrzędnych dwuwymiarowych do drugiego opisują proste wzory.



Rys. 1.2.5. Układ współrzędnych biegunowych

Konwersja współrzędnych biegunowych punktu do współrzędnych kartezjańskich:

$$x = r \cdot \cos \theta$$

$$y = r \cdot \sin \theta$$

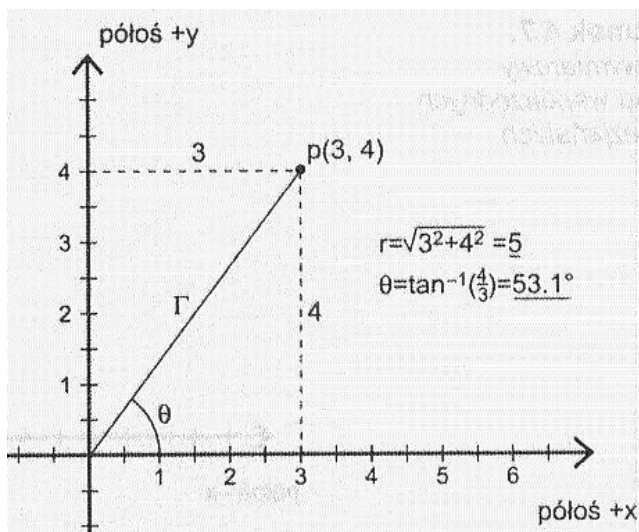
Konwersja współrzędnych kartezjańskich punktu do współrzędnych biegunowych jest bardziej złożona:

$$x^2 + y^2 = r^2$$

$$r = \sqrt{x^2 + y^2}$$

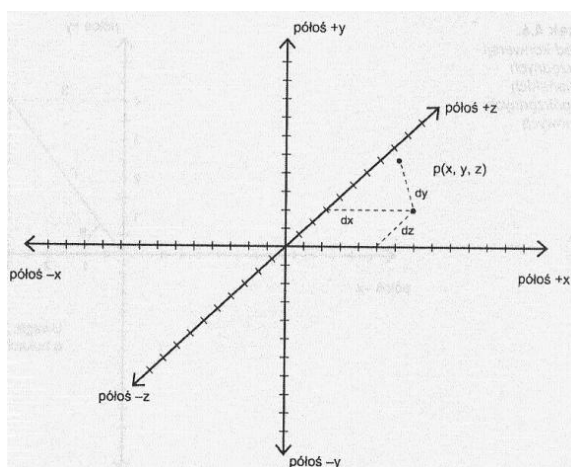
$$\theta = \tan^{-1} \left( \frac{y}{x} \right)$$

Na rysunku 1.2.6. pokazane są wzajemne relacje pomiędzy współrzędnymi punktu P w układzie kartezjańskim i biegunowym.



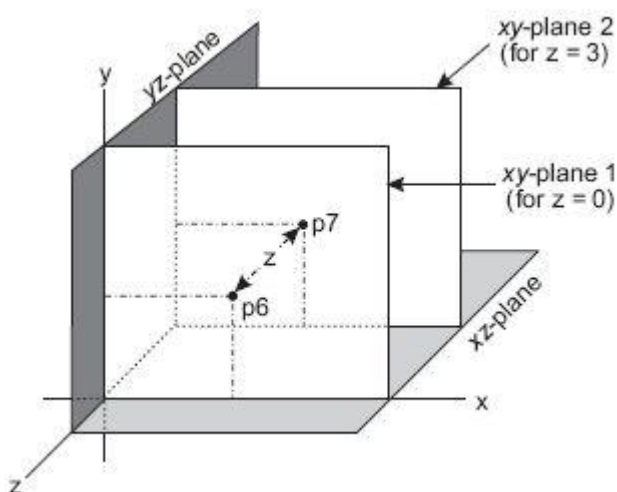
Rys. 1.2.6 Konwersja współrzędnych punktu P

Trójwymiarowy układ współrzędnych kartezjańskich jest rozszerzeniem układu dwuwymiarowego przez dodanie trzeciej osi – osi z, prostopadłej do osi x i osi y. W trójwymiarowym układzie współrzędnych lokalizację punktu (rys. 1.2.7) określają trzy liczby. Dla punktu oznaczonego jako P(1,2,3), współrzędna x ma wartość 1, współrzędna y ma wartość 2 a współrzędna z ma wartość 3. Wyróżniamy dwa układy współrzędnych trójwymiarowych – układ prawoskrętny i lewoskrętny, typ układu zależy od ustalenia dodatniego kierunku osi z.



Rys.1.2.7. Trójwymiarowy układ współrzędnych (układ lewoskrętny)

Trzy osie układu współrzędnych wyznaczają trzy płaszczyzny – płaszczyznę  $xy$ , płaszczyznę  $xz$  i płaszczyznę  $yz$ . Te płaszczyzny są do siebie wzajemnie prostopadłe. Lokalizację punktu  $P$  w przestrzeni trójwymiarowej możemy wizualizować przy pomocy lokalizacji punktu na płaszczyźnie  $xy$  i umieszczając te płaszczyzny kolejno na osi  $z$  (rys. 1.2.8)



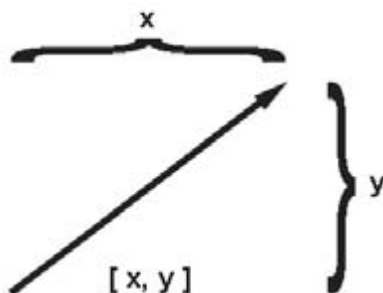
Rys.1.2.8. Trójwymiarowy układ współrzędnych, płaszczyzny  $xy$ ,  $xz$  i  $yz$

Punkt  $p_6$  umieszczony na pierwszej płaszczyźnie  $xy$  został odwzorowany na drugiej płaszczyźnie  $xy$  jako  $p_7$ . W ten sposób można szybko rysować obiekty trójwymiarowe, takie jak na przykład prostopadłościany.



Dość często, szczególnie w modelowaniu posługujemy się innymi układami współrzędnych trójwymiarowych. W fizyce popularne są układy takie jak układ współrzędnych cylindrycznych i układ współrzędnych sferycznych.

W geometrii analitycznej najważniejszym pojęciem jest wektor. W zasadzie większość algorytmów grafiki trójwymiarowej wykorzystuje pojęcie wektora. W fizyce pojęcie wektora odgrywa zasadniczą rolę. Formalnie wektor jest to uporządkowany zbiór liczb (para liczb dla wektora w płaszczyźnie, trójka w przestrzeni 3D). Wielkość wektorowa to wielkość, która ma zarówno wartość (wartość bezwzględna, moduł), jak i kierunek. Działania na wektorach podlegają pewnym, ściśle określonym prawom. Jest to tzw. rachunek wektorowy. Każdy programista gier komputerowych powinien posiadać elementarną wiedzę w zakresie wykonywania operacji na wektorach. Matematycy wyróżniają wektory i skalary. Skalar oznacza po prostu liczbę. Używamy tej nazwy, aby podkreślić, że jakaś wielkość nie jest wektorem. Na przykład w grafice i fizyce „prędkość” i „przesunięcie” są wielkościami wektorowymi, a „szybkość” i „odległość” są wielkościami skalarnymi. W interpretacji geometrycznej wektor jest skierowanym odcinkiem (rys. 1.2.9), który ma długość (ang. *magnitude*) i kierunek (ang. *direction*).



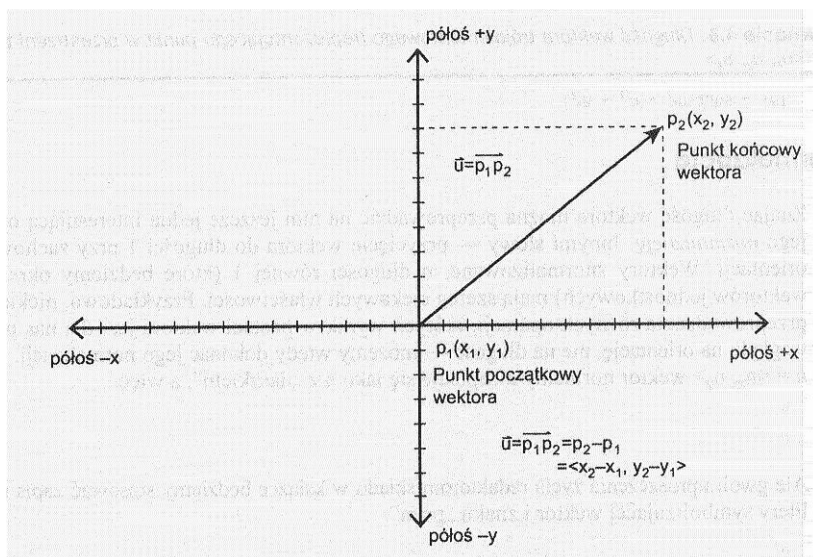
Rys.1.2.9. Wizualizacja wektora dwuwymiarowego

Ponieważ wektor nie ma położenia, możemy go przesuwac w dowolne miejsce. Dla celów poglądowych umieścimy rozważany wektor w początku układu współrzędnych (rys.1.2.10)

Odcinek skierowany wyznaczony jest punktem początkowym  $p_1$  i punktem końcowym  $p_2$ . Wektor  $u(a,b)$  reprezentuje odcinek pomiędzy punktami  $p_1(x_1,y_1)$  i  $p_2(x_2,y_2)$ . Składowe wektora  $u$  wyliczamy zgodnie z wzorami:

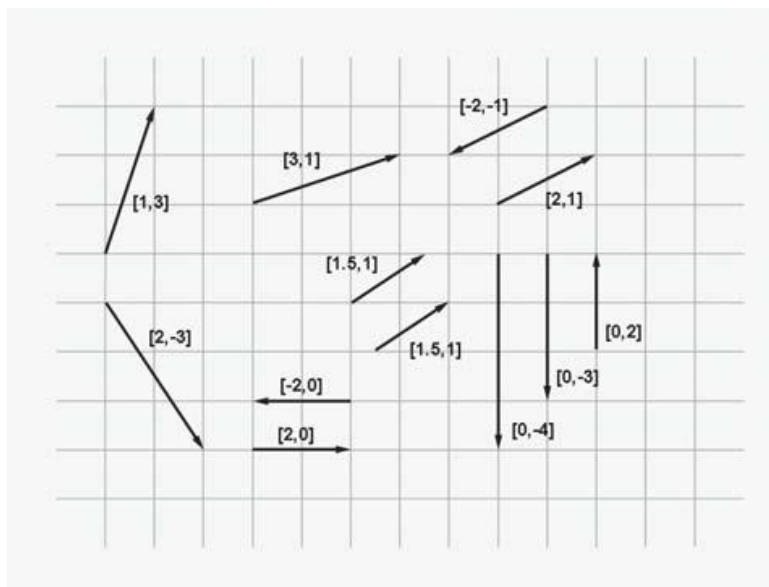
$$a = x_2 - x_1$$

$$b = y_2 - y_1$$



Rys.1.2.10. Wektor dwuwymiarowy i jego składowe.

Wektory są bardzo użyteczne w reprezentowaniu wielkości fizycznych takich jak na przykład prędkość, przyspieszenie, siła tarcia, itp. Wektory mogą mieć różne długości i różne kierunki. Na rysunku 1.2.11 pokazano wektory o różnych składowych.

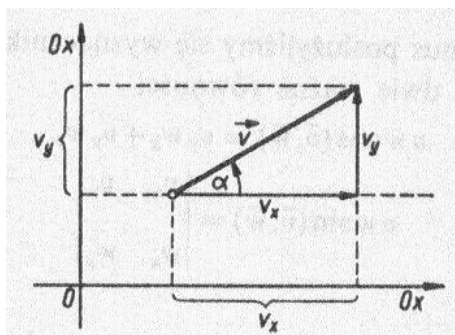


Rys.1.2.11. Przykłady wektorów dwuwymiarowych.

Moduł (długość) wektora  $u(a,b)$  wyliczamy na podstawie związku Pitagorasa:

$$u^2 = a^2 + b^2$$

Składową wektora nazywamy rzut wektora na wybraną oś współrzędnych, na rysunku 1.2.12  $v_x$  jest składową wektora  $v$  wzdłuż osi  $x$ ,  $v_y$  jest składową wektora  $v$  wzdłuż osi  $y$ .



Rys. 1.2.12. Składowe  $v_x$  i  $v_y$  wektora  $v$

W naszym przykładzie obie składowe mają wartość dodatnią. Składową wektora można obliczyć następująco:

$$\begin{aligned} v_x &= v \cos \theta \\ v_y &= v \sin \theta \end{aligned}$$

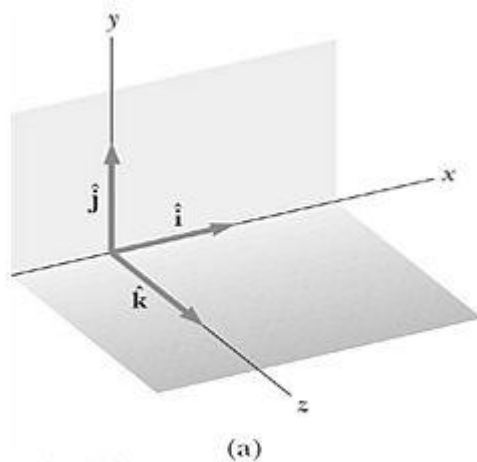
Wektor  $v$  może być reprezentowany jako suma jego składowych  $v_x$  i  $v_y$  (rys. 1.2.12):

$$\begin{aligned} v &= v_x + v_y \\ \alpha &= \tan^{-1} \frac{v_y}{v_x} \end{aligned}$$

Bardzo użyteczną do operowania wektorami jest koncepcja wektorów jednostkowych (wersorów). Wektorem jednostkowym nazywamy wektor o długości równej 1, skierowany w określonym kierunku. Wersor nie ma wymiaru ani jednostki, służy do wyznaczania kierunku w przestrzeni 3D, wersory oznaczamy jako  $\mathbf{i}$ ,  $\mathbf{j}$ ,  $\mathbf{k}$ . Wektory jednostkowe przedstawione są na rysunku 1.2.13. Wersorów można użyć do zapisu wektorów. Dwuwymiarowy wektor  $a$  można zapisać w postaci:

$$a = a_x \mathbf{i} + a_y \mathbf{j}$$

Wielkości  $a_x$  oraz  $a_y$  są wektorami składowymi wektora  $a$ .

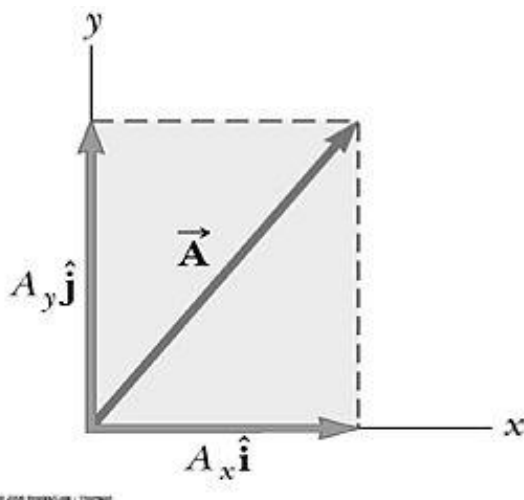


Rys.1.2.13. Wektory jednostkowe (wersory)  $\hat{i}$ ,  $\hat{j}$  i  $\hat{k}$

Dowolny wektor, przy wykorzystaniu wersorów może być przedstawiony w następującej postaci ( wektor 3D):

$$A = A_x \hat{i} + A_y \hat{j} + A_z \hat{k}$$

Na rys. 1.2.14 przedstawiony jest dwuwymiarowy wektor  $\vec{A}$ .



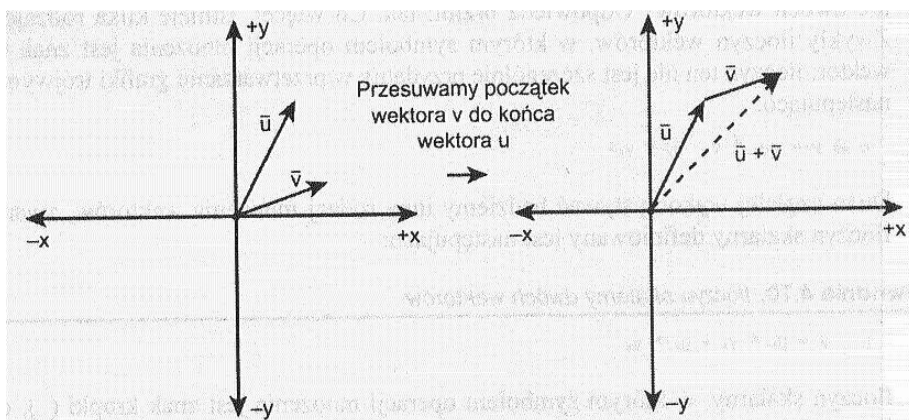
Rys. 1.2.14 Dwuwymiarowy wektor  $\vec{A}$ , zaznaczone są wersory

Działania na wektorach podlegają ściśle określonym prawom – prawom rachunku wektorowego.

Na wektorach można wykonać następujące operacje:

- dodawanie wektorów
- odejmowanie wektorów
- mnożenie wektora przez skalar
- mnożenie - iloczyn skalarny
- mnożenie - iloczyn wektorowy

Dodając wektory należy pamiętać, że możemy dodawać wektory reprezentujące te same wielkości (np. nie można dodawać wektora prędkości do wektora przyspieszenia). Wektory możemy dodawać metodą geometryczną lub analityczną. W metodzie geometrycznej wykonuje się odpowiedni rysunek, w ustalonym układzie współrzędnych nanosi się wektory, łączy się koniec pierwszego wektora z początkiem drugiego (gdy dodaje się dwa wektory) i rysuje się wektor łączący początek pierwszego wektora z końcem drugiego wektora (rys.1.2.14)



Rys. 1.2.14 Dodawanie geometryczne dwóch wektorów

W metodzie analitycznej w celu dodania dwóch wektorów (lub ich większej ilości) należy zsumować odpowiednie składowe wektorów. Dodawanie dwóch wektorów  $a$  i  $b$

$$a + b = r$$

korzystając z wersorów realizujemy następująco:

$$r = (a_x i + a_y j) + (b_x i + b_y j)$$

$$r = (a_x + b_x) i + (a_y + b_y) j$$

Wobec tego mamy:

$$\begin{aligned} r_x &= a_x + b_x \\ r_y &= a_y + b_y \end{aligned}$$

oraz:

$$\begin{aligned} |r| &= \sqrt{r_x^2 + r_y^2} \\ \theta &= \tan^{-1} \frac{r_y}{r_x} \end{aligned}$$

Iloczyn skalarny dwóch wektorów  $a$  i  $b$  jest to liczba równa iloczynowi modułów tych wektorów i cosinusa kąta między nimi:

$$a \cdot b = |a||b| \cos(a, b)$$

Iloczyn skalarny dwóch wektorów równa się sumie iloczynów składowych tych wektorów:

$$a \cdot b = a_x b_x + a_y b_y$$

Jeżeli dwa wektory  $a$  i  $b$  są prostopadłe do siebie, to ich iloczyn skalarny jest równy zeru:

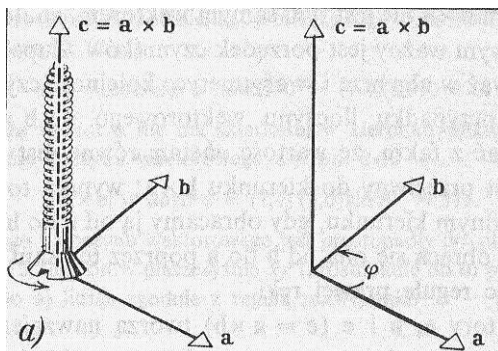
$$a_x b_x + a_y b_y = 0$$

Jeżeli dwa wektory  $a$  i  $b$  są równoległe to wyznacznik tych wektorów jest równy zeru:

$$\begin{vmatrix} a_x & a_y \\ b_x & b_y \end{vmatrix} = 0$$

Iloczyn wektorowy dwóch wektorów  $a$  i  $b$  jest to wektor  $c$  o długości równej iloczynowi modułów tych wektorów i sinusa kąta między nimi:

$$a \cdot b = |a||b| \sin(a, b)$$



Rys. 1.2.15. Iloczyn wektorowy

Kierunek wektor  $c$  jest prostopadły do płaszczyzny utworzonej przez dwa wektory  $a$  i  $b$ . Dodatni kierunek wektora  $c$  określamy odwołując się do reguły śruby prawoskrętnej (rys. 1.2.15).

Wektory mają wiele zastosowań w fizyce. Użyteczność wektorów w opisie zjawisk fizycznych wynika z faktu, że związki między wektorami są niezmiennicze względem translacji i obrotów układu współrzędnych. W klasycznej fizyce prawa fizyki także nie zmieniają się podczas translacji i obrotów układu współrzędnych. Notacja wektorowa jest idealna do wyrażania praw fizyki.

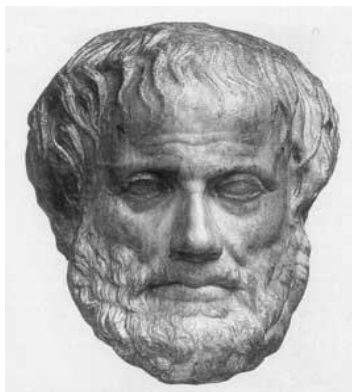
### 1.3. Siła i zasady dynamiki Newtona

W realistycznych grach komputerowych chcemy aby obiekty poruszały się w sposób naturalny, zgodny z naszym doświadczeniem. Musimy mieć narzędzia pozwalające nam w sposób wystarczająco precyzyjny opisać ruch obiektów.

Musimy brać pod uwagę uwarunkowania (ograniczenia) sprzętowe. Przetwarzanie komputerowe nie może być zbyt kosztowne. Musimy wyznaczyć granice dokładności modelowania i symulacji (korzystając na przykład z rachunku różniczkowego).

Poglądy na zagadnienie ruchu (mechanikę) zmieniały się na przestrzeni dziejów. Największy wkład w wyjaśnienie ruchu wnieśli trzej znakomici uczeni – Arystoteles, Galileusz i Isaac Newton.

Znakomity filozof starożytny, Arystoteles (384-322 p.n.e) urodził się w Stagirze na półwyspie Chalcydyckim



Rys. 1.3.1 Arystoteles (384 – 322 p.n.e)

Wychował się w stolicy Macedonii. W wieku 17 lat udaje się do Aten, gdzie spędzi 20 lat, będąc uczniem Akademii Platńskiej. W latach 343-340 p.n.e. jest nauczycielem Aleksandra Macedońskiego. W 335 r. p.n.e. zakłada w Atenach uczelnię i centrum, badawcze - słynne Liceum.

Arystoteles jest uważany za najwszechstronniejszego i największego uczonego starożytności. Jego system wiedzy, tłumaczący wszystkie aspekty świata ożywionego i nieożywionego przetrwał w niezminionej postaci prawie dwa tysiące lat. Napisał osiem ksiąg Fizyki. Najważniejszym zagadnieniem badanym przez Arystotelesa był ruch. Mówił: „nieznajomość istoty ruchu mogłaby doprowadzić w konsekwencji do nieznajomości przyrody”. Analizowany przez Arystotelesa ruch wymagał przyczyny.

Prawa mechaniki arystotelesowskiej są następujące:

- Ciało nie poddane wpływom zewnętrznym znajduje się w spoczynku
- Prędkość ciała wprawionego w ruch przez zewnętrzną przyczynę jest proporcjonalna do działającej siły i odwrotnie proporcjonalna do oporu, jaki stawia ośrodek

Arystoteles udowadniał niemożliwość istnienia próżni. Ani on, ani Galileusz czy Newton nie znali pojęcia prędkości, jako stosunku drogi do czasu! Pojęcie to wprowadził Euler w XVIII wieku.

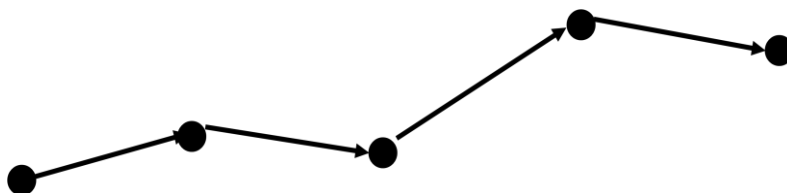
Arystoteles przyjmował następujące argumenty obserwacyjne:

- Jeżeli przesuwany jest obiekt, to mamy ruch, jeżeli przestaniemy przesuwać obiekt, to obiekt zatrzyma się
- Kamień spada szybciej niż piórko

Na podstawie tych obserwacji mamy wnioski:

- Obiekty chcą się zatrzymywać
- Ruch jest liniowy (trajektoria jest linią prostą)
- Ruch występuje jedynie, gdy jest przyczyna
- Obiekty cięższe spadają szybciej

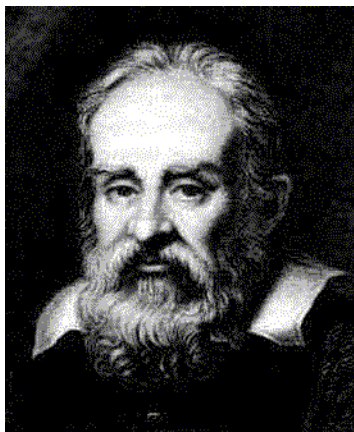
Barry Casper napisał (1977): „mamy tylko dwie możliwości: albo Arystoteles był głupcem, albo miał, co innego na myśli. Ponieważ istnieje dostatecznie dużo dowodów, że głupcem nie był, staraliśmy się odkryć możliwą do przyjęcia odmienną interpretację jego słów.



Rys1.3.2. Według Arystotelesa ruch jest zmianą położenia



Twórca nowoczesnej fizyki Galileo Galilei (1564 – 1642), zwany w Polsce Galileuszem, urodził się w Pizie. W samym roku (1564 ) umiera Michał Anioł oraz rodzi się Szekspir.



Rys. 1.3.3. Galileusz (1564 – 1642)

W wieku 17 lat rozpoczyna studia medyczne na uniwersytecie w Pizie. Bardziej interesuje się matematyką i mechaniką, przerywa studia, nie uzyskuje dyplomu, żyje z korepetycji. W 1589 otrzymuje posadę profesora matematyki na uniwersytecie w Pizie. Jest wynalazcą wagi hydrostatycznej, publikuje traktat o środku ciężkości (1589 r.) Około 1590 r. opracowuje traktat *De Motu*, w którym krytykuje teorię ruchu Arystotelesa. W 1592 r. otrzymuje katedrę matematyki na uniwersytecie w Padwie. W 1604 r. odkrył prawo swobodnego spadku. W 1609 roku demonstruje zbudowaną przez siebie lunetę. Dzięki nowemu przyrządowi odkrywa góry na Księżycu, satelity Jowisza, miliardy gwiazd Drogi Mlecznej. Obserwacje publikuje w traktacie *Sidereus nuncius* (Gwiezdny zwiastun, Wyślanek gwiazd, Posłanie z gwiazd) w 1610 r. Około 500 egzemplarzy wykupiono natychmiast, traktat był tłumaczony na wiele języków, w 1615 r. ukazał się przekład chiński. Wynaleziona luneta stała się przebojem rynkowym, jedną z lunet otrzymał król polski Władysław. 5 marca 1616 r. został wydany dekret potępiający kopernikanizm. W 1632 r. ukazał się „Dialog o dwu najważniejszych układach świata, ptolemeuszowym i kopernikowym”, wydany za przyzwoleniem cenzury. W 1632 przeciwnicy Galileusza wszczynają kampanię zwalczającą jego poglądy, papież Urban VIII wydaje zgodę na proces Galileusza. 12 kwietnia 1633 r. rozpoczyna się proces Galileusza, oskarżono go o naruszenie zakazu dyskusowania o systemie Kopernika. Galileusz odwołał swoje poglądy. W areszcie domowym pisze dzieło omawiające jego badania w zakresie fizyki. Dzieło, przemycone do Holandii i tam wydane (1638 r.) miało tytuł: „*Discorsi e dimostrazioni intorno a due nuove scienze attenenti alla Mechanica i Movimenti*” (Rozmowy i dowodzenia matematyczne w zakresie dwóch nowych umiejętności dotyczących mechaniki i

ruchów miejscowych).

Najważniejsze obserwacje Galileusza to:

- gdy nie ma oporu powietrza, wszystkie ciała spadają z jednakową prędkością
- w spadku swobodnym ciało porusza się ze stałym przyspieszeniem, oznacza to, że jego prędkość jest proporcjonalna do czasu
- droga przebyta przez ciało spadające jest proporcjonalna do kwadratu czasu
- tor ruchu pocisków i ciał w rzucie ukośnym jest parabolą

Ważne osiągnięcia Galileusza w zakresie mechaniki to:

- ustalił zasadę względności ruchu
- ustalił prawo swobodnego spadku ciał
- dowiódł, że tor pocisku jest parabolą
- podał zasady bezwładności

Nowoczesną fizykę, nie zmienioną do dzisiaj, zapoczątkował angielski uczoney Isaac Newton (1642 – 1727), który urodził się w Woolsthorpe (Lincolnshire) w Anglii.



Rys.1.3.4 Isaac Newton (1642 – 1727)

Isaac Newton ma ogromne osiągnięcia w fizyce, jest jednym z najznakomitszych matematyków wszystkich czasów, odkrył (niezależnie od Leibniza) rachunek różniczkowy i całkowy, który nazywał rachunkiem fluksji i fluent. W 1661 r. został przyjęty do Trinity College w Cambridge (w zamian za usługiwanie zamożniejszym studentom, mógł sam studiować). Studiował grekę, logikę, etykę i retorykę. W 1663 utworzono w Cambridge katedrę Lucasa (nauka matematyki), wykłady zaczęły się w 1664 r. Matematykę wykładał Izaak Barrow.

W 1665 Newton uzyskuje stopień bakałarza, w 1668 stopień magistra (ang. *master of arts*). W 1669 r. z rekomendacji Barrowa, Newton zostaje jego następcą w katedrze Lukasa i przez około 20 lat wykłada matematykę, mechanikę i optykę.

5 lipca 1687 r. fundamentalne dzieło Newtona „*Philosophiae naturalis principia mathematica*” (Zasady matematyczne filozofii naturalnej) zostało wydrukowane. Pierwotnie Newton chciał napisać dwie księgi o tytule *De Motu Corporum* (O ruchu ciał), ostatecznie napisał trzy księgi.

Pierwsza księga Zasad liczy 14 rozdziałów:

- Rozdział 1 zawiera lematy matematyczne
- Rozdziały 2-7 są poświęcone opisowi orbit poruszających się ciał pod wpływem siły centralnej
- Rozdziały 8-13 są poświęcone skomplikowanym zagadnieniom: przyciąganie ciał sferycznych i niesferycznych, ruch trzech ciał,
- Rozdział 14 omawia podstawy optyki korpuskularnej (np. prawo załamania)

Druga księga Zasad liczy 9 rozdziałów, omawiane są:

- Ruch ciał w ośrodkach stawiających opór (hydrodynamika)
- Ruch wahadła w ośrodku
- Uwagi do teorii wirów Kartezjusza
- Omówienia prawa Keplera

Trzecia księga Zasad omawia zastosowania teorii Newtona do wyjaśnienia ruchu planet, satelitów i komet, w tej księdze sformułowane jest też prawo ciężenia powszechnego. W tej księdze znajdują się między innymi:

- Dowód, że komety poruszają się po krzywych stożkowych
- Metoda wyznaczania orbit komet,
- Dowód na proporcjonalność masy bezwładnej i grawitacyjnej,
- Dyskusja na temat spłaszczenia Ziemi na skutek ruchu obrotowego
- Omówienie przyptywu i odpływu mórz spowodowane przyciąganiem Księżyca i Słońca
- Omówienie zjawiska precesji

Na początku tej księgi znajdują się **Prawidła badania natury**, jej postać to:

**„Nie należy dla zjawisk natury przypuszczać więcej przyczyn, niż te, które są prawdziwe i wystarczają do ich objaśnienia”.**

Prawa ruchu sformułowane przez Newtona są podstawą nauki o mechanice, zwane także pod nazwą trzech zasad dynamiki Newtona.

Wykorzystując te zasady tworzymy realistyczne animacje.

Zasady dynamiki omawia Newton w wydany w 1687 roku dziele „*Philosophiae Naturalis Principia Mathematica*”. Prawa ruchu mają następującą postać:

1. Każde ciało trwa w swym stanie spoczynku lub ruchu prostoliniowego i jednostajnego, jeśli siły przyłożone nie zmuszają ciała do zmiany tego stanu.
2. Zmiana ruchu jest proporcjonalna do przyłożonej siły poruszającej i odbywa się w kierunku prostej, wzdłuż której siła jest przyłożona (F- siła, m- masa, a- przyspieszenie)

$$F = ma$$

1. Względem każdego działania istnieje przeciwdziałanie, skierowane przeciwnie i równe; tj. wzajemne działania dwóch ciał są zawsze równe sobie i skierowane przeciwnie

W pierwszej zasadzie mamy pytanie: względem, jakiego układu odniesienia ustala się spoczynek lub ruch. Z obserwacji wiemy, że pierwsza zasada nie jest słuszna względem wszystkich układów odniesienia. Rozpatrzmy pasażera w wagonie. Gdy pociąg jedzie spokojnie, pierwsza zasada jest spełniona. W momencie, gdy wagon hamuje, lub wchodzi w zakręt, obiekty w wagonie zaczynają się poruszać bez widocznego oddziaływania otaczających go obiektów. Mamy naruszenie pierwszej zasady. Układ odniesienia, do którego stosuje się pierwsza zasada, nazywa się **układem inercyjnym**. Pierwsza zasada nazywa się inaczej **zasadą bezwładności** (inercji). Wystarczająco dokładnym układem inercyjnym jest układ heliocentryczny, układem inercyjnym będzie też każdy układ poruszający się względem układu heliocentrycznego jednostajnie i prostoliniowo. Każdy układ mający względem układu inercyjnego przyspieszenie, nie będzie już układem inercyjnym.

Druga zasada wprowadza pojęcie siły. Wielkość fizyczna – siła charakteryzuje działanie jednych ciał na drugie, w wyniku, którego ciała nabywają przyspieszenia. Ze wzoru wynika, że im silniejsze działanie, tym większego przyspieszenia ciało nabywa. Przy jednoczesnym działaniu kilku sił ciało nabywa takiego przyspieszenia, jakiego udziela mu siła, będąca sumą geometryczną danych sił. Siła jest wektorem. Wektor siły jest skierowany tak samo jak wektor wywołanego przez nią przyspieszenia. Im mniejszego przyspieszenia nabywa ciało pod działaniem danej siły, tym większa jest jego masa. Masy różnych ciał są odwrotnie proporcjonalne do przyspieszeń nadawanych tym ciałom przez równe siły.

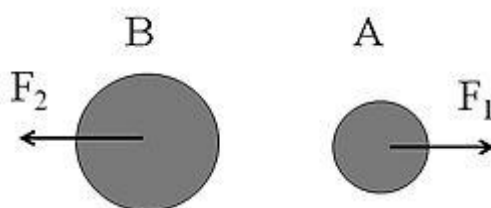
$$\frac{m_1}{m_2} = \left| \frac{a_2}{a_1} \right|$$

Trzecia zasada Newtona uzupełnia drugą zasadę Newtona.

Jeśli ciało B działa na ciało A z siłą  $F_1$ , to ciało A działa z kolei na ciało B z siłą  $F_2$ , liczbowo równą sile  $F_1$  i skierowana przeciwnie:

$$\vec{F}_1 = -\vec{F}_2$$

Rysunek ilustruje trzecią zasadę Newtona, pokazane jest oddziaływanie dwóch kul.



Rys. 1.3. 5. Trzecia zasada Newtona

W nowoczesnym ujęciu zasady dynamiki Newtona mają następującą postać (definicje wg. Hollidaya, Resnicka i Walkera):

1. Jeśli na ciało nie działa żadna siła, to nie może zmienić się jego prędkość, czyli nie może ono przyspieszyć.
2. Siła wypadkowa działająca na ciało jest równa iloczynowi masy tego ciała i jego przyspieszenia.
3. Gdy dwa ciała oddziałują ze sobą, siły jakimi działają one na siebie mają taką samą wartość bezwzględną i przeciwne kierunki

W przyrodzie występuje wiele rodzajów sił. W realistycznej grafice komputerowej ważną rolę odgrywają siły: siła ciężkości, siła tarcia oraz oporu.

**Siła ciężkości** (grawitacja)  $F_g$  jest to siła, z jaką dane ciało jest przyciągane przez inne ciało. Praktycznie rozważamy siły, z jakimi inne ciała przyciągane są przez Ziemię. Siła ciężkości jest skierowana do środka Ziemi, pionowo w dół. Zazwyczaj przyjmujemy, że mamy do czynienia ze spadkiem swobodnym ciała o masie  $m$  z przyspieszeniem ziemskim o wartości  $g$  ( $= 9.81 \text{ m/sek}^2$ ).

Zgodnie z drugą zasadą dynamiki Newtona mamy:

$$F_g = mg$$

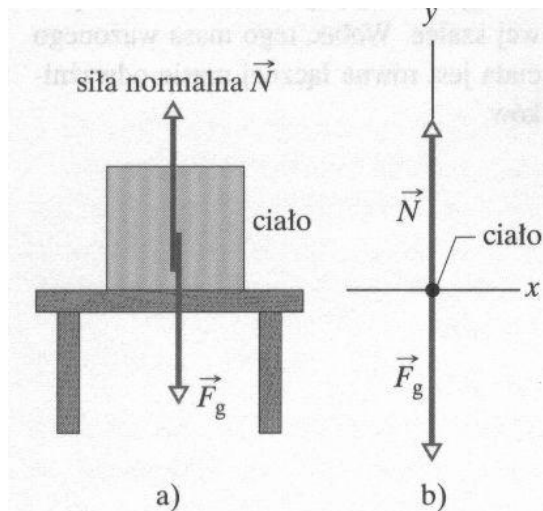
Ciężarem  $W$  nazywamy wartość bezwzględną siły potrzebnej do zapobieżenia spadkowi ciała, mierzonej przez obserwatora na Ziemi.

$$W = mg$$

Pomiar ciężaru ciała wykonujemy, gdy ciało nie porusza się z przyspieszeniem pionowym w stosunku do Ziemi. Pomiar wykonany na przykład w łazience jest prawidłowy, wykonany w poruszającej się windzie da inny wynik (ciężar pozorny).

Ciężar ciała to inna wielkość fizyczna niż jego masa. Ciężar to wartość siły, a masa jest opisana we wzorze  $W=mg$ . Gdy znajdziemy się na Księżycu, to masa ciała się nie zmieni, ale ponieważ  $g$  jest inne, to ciężar się zmieni! Na Księżycu przyspieszenie wynosi  $1.7 \text{ m/sek}^2$ .

Gdy stoimy na podłodze, Ziemia działa na nas siłą ciężkości skierowaną pionowo w dół, mimo działania tej siły nie poruszamy się. Jesteśmy w spoczynku, ponieważ podłoga, która ugina się pod nami, działa na nas siłą skierowaną ku górze. Siła, którą działa na nas podłoga (a także inne tego typu ciała) nazywamy **siłą normalną** i oznaczamy przez  $N$ .



Rys. 1.3.6. Siła ciężkości i siła normalna, ciało jest w spoczynku

Drugą zasadę dynamiki Newtona możemy zapisać w postaci:

$$N - F_g = ma_y$$

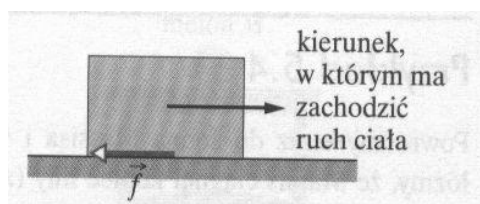
$$N - mg = ma_y$$

$$N = mg + ma_y = m(g + a_y)$$

Ostatni wzór obowiązuje dla dowolnej wartości przyspieszenia stołu i klocka w kierunku pionowym (np. gdy mamy stół w poruszającej się windzie). Jeżeli winda stoi to mamy:

$$N = mg$$

Na ciało będące w ruchu (np. samochód na jezdni) działają siły w wyniku oddziaływania między ciałem i powierzchnią. Przyjmuje się, że opory ruchu można opisać za pomocą jednej siły  $\mathbf{f}$ , nazywanej **siłą tarcia**, lub krótko – **tarciem**. Siła tarcia skierowana jest wzdłuż powierzchni, przeciwnie do kierunku, w którym zachodzi ruch ciała.



Rys.1.3.7. Siła tarcia

Jeżeli zachodzi ruch ciała w cieczy lub w gazie to na ciało działa **siła oporu**. Ta siła jest skierowana w kierunku przepływu ciała i utrudnia ruch. Wyznaczenie tej siły jest dość skomplikowany zagadnieniem. Jeżeli rozpatrujemy ruch piłki w powietrzu, przy założeniu, że ruch jest na tyle szybki, aby przepływ powietrza był turbulentny (powstają wiry) to wartość siły oporu  $D$  dana jest wzorem:

$$D = \frac{1}{2} C \rho S v^2$$

W tym wzorze  $C$  oznacza współczynnik oporu (wyznaczany doświadczalnie),  $\rho$  - oznacza gęstość powietrza,  $S$  – jest polem przekroju poprzecznego ciała,  $v$  jest prędkością. Współczynnik oporu  $C$  jest zazwyczaj zawarty w przedziale (0.4, 1.0).

Jeżeli ciało o masie  $m$  spada w powietrzu (działa siła grawitacji  $F_g$ ) to siła oporu skierowana  $D$  jest do góry, i jej wartość wzrasta w miarę jak wzrasta prędkość ciała  $v$ . Równanie sił ma postać:

$$D - F_g = ma$$

Gdy ruch trwa dostatecznie długo, siły oporu i grawitacji zrównoważą się, co oznacza, że przyspieszenie  $a = 0$ .

W tym momencie ciało porusza się ze stałą prędkością. Ta prędkość nosi

nazwę prędkości granicznej. Z przytoczonych wzorów możemy wyznaczyć prędkość graniczną:

$$v_{gr} = \sqrt{\frac{2F_g}{C\rho S}}$$

W tabeli 1.2 umieszczone są wybrane prędkości graniczne (z podręcznika D.Holliday, R. Resnick, J. Walker).

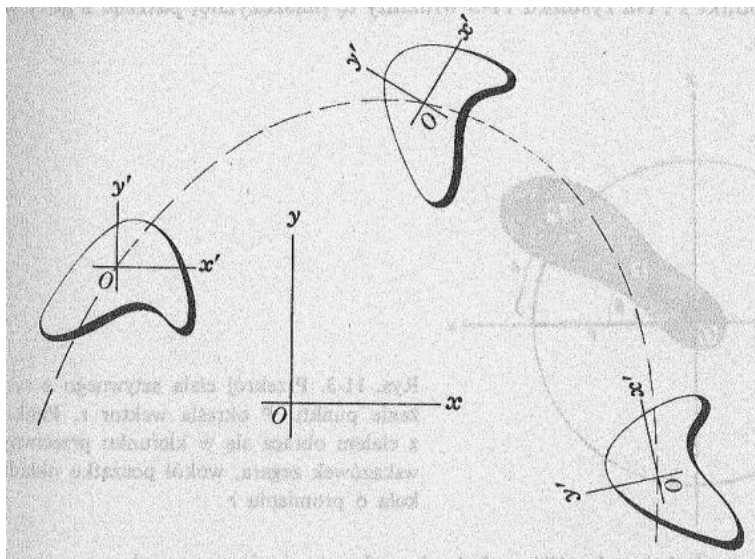
Tabela 1.2. Prędkości graniczne

Ciało	$V_g$ (m/s)
Skoczek (bez spadochronu)	60
Skoczek (ze spadochronem)	5
Piłka baseballowa	42
Piłka tenisowa	31
Piłka pingpongowa	9
Kropla deszczu	7

#### 1.4. Punkt materialny, środek masy, moment bezwładności

W grach komputerowych modelujemy ruch obiektów. Mechanika, jeden z najstarszych działów fizyki zajmuje się badaniem i opisem ruchów ciał. Dział mechaniki zajmujący się opisem samego ruchu nosi nazwę kinematyka. Opisem ruchu pod działaniem sił zajmuje się dział fizyki zwany dynamiką. Ruch może być postępowy oraz obrotowy. Przykładem ruchu będącego jednocześnie ruchem postępowym i obrotowym jest ruch rzuconej piłki wirującej wokół swojej osi. Na rysunku 1.2.16 przedstawiono przypadek ruchu ciała sztywnego. Ruch tego ciała złożony jest z ruchu postępowego i obrotowego.





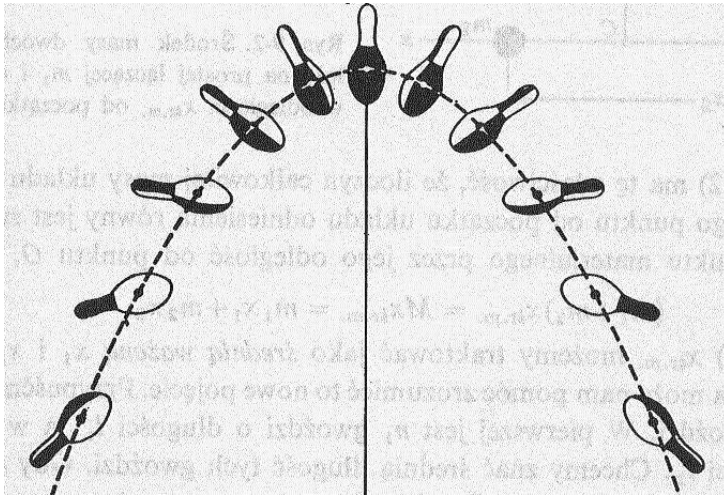
Rys. 1.2.16. Postępowo-obrotowy ruch ciała sztywnego.

W opisie i modelowaniu ruchu wygodnym jest pojęcie punktu materialnego. Punktem materialnym nazywamy ciało, które ma masę, ale nie ma objętości. Oczywiście takie ciało nie występuje w przyrodzie, natomiast w opisie zjawisk fizycznych znacznie upraszcza wszelkie rozważania. Ciało możemy traktować, jako punkt materialny, jeżeli jego rozmiary są bardzo małe w porównaniu z przestrzenią, w jakiej zachodzi konkretne zjawisko. Klasycznym przykładem jest opis ruchu planety Ziemi wokół Słońca. Jeżeli rozważymy odległości Ziemia- Słońce, to rozmiar zarówno Ziemi jak i Słońca w porównaniu do tej odległości jest bardzo mały, tak więc potraktowanie Ziemi, jako punktu materialnego jest bardzo dobrze uzasadnione. Jeżeli ciało sztywne porusza się jedynie ruchem postępowym, zawsze traktujemy je, jako punkt materialny (chyba, że rozważać będziemy kolizje). Bryła sztywna składa się z wielu punktów materialnych, ale w ruchu postępowym wszystkie przemieszczają się jednakowo, mając opis ruchu jednego punktu, mamy opis wszystkich punktów wchodzących w skład opisywanego ciała.

Kolejnym użytecznym pojęciem jest **środek masy**. Jeżeli wyrzucimy do góry kij baseballowy lub maczugę gimnastyczną, to ruch tego obiektu będzie bardzo skomplikowany, ponieważ będzie złożeniem ruchu postępowego i obrotowego. Każda część kija porusza się inaczej. Mówimy, że mamy do czynienia z ruchem punktów materialnych. Analiza zjawiska wskazuje, że zagadnienie opisu ruchu takiego ciała można znacznie uprościć (nic nie tracąc z precyzji i dokładności) Jeżeli zauważymy, że istnieje taki punkt kija, który porusza się po paraboli, dokładnie tak samo jak wyrzucony do góry obiekt, to ten punkt może być przybliżonym punktem materialnym. Ten punkt nazywamy środkiem masy.

Formalnie:

*środek masy ciała lub układu ciał to punkt, który porusza się tak jakby cała masa ciała była umieszczona w tym punkcie, wszystkie siły działające na to ciało były przyłożone do tego punktu.*

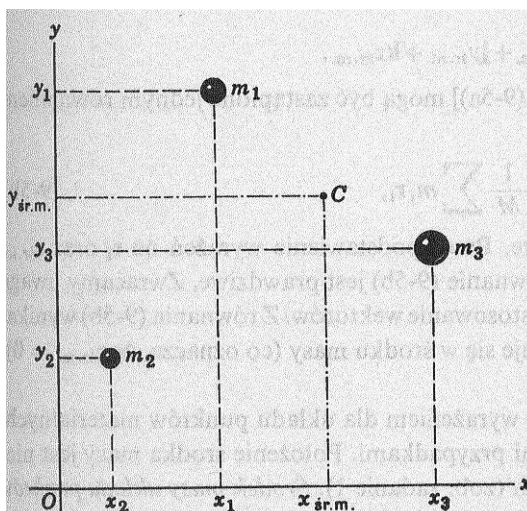


Rys. 1.2.17 Rzucona w powietrze maczuga gimnastyczna. Zaznaczony kropką środek masy porusza się po paraboli (tak jak rzucona np. piłka) a wszystkie inne punkty maczugi poruszają się skomplikowanym ruchem postępowo-obrotowym.

Gdy siłą zewnętrzną jest siła ciężkości, wtedy działa ona na środek ciężkości. W jednorodnym polu grawitacyjnym, środek ciężkości pokrywa się ze środkiem masy. Pojęcie środka masy jest pojęciem bardziej ogólnym.

Dla jednorodnych płaskich figur takich jak trójkąt, prostokąt czy okrąg wyznaczenie środka masy jest intuicyjne – w trójkącie jest to punkt przecięcia się środkowych boków trójkąta, w prostokącie jest to punkt przecięcia się przekątnych prostokąta, w okręgu jest to po prostu środek okręgu. Dla wielu ciał i układu ciał także stosunkowo prosto można wyznaczyć (obliczyć) środek masy. W przypadku skomplikowanych kształtów w celu wyznaczenia środka masy musimy zastosować rachunek całkowy, aczkolwiek w grafice komputerowej rzadko wykorzystujemy całkowanie, raczej całkowanie zastępujemy sumowaniem.

Rozważmy najpierw przypadek trzech punktów materialnych o różnych masach leżących na płaszczyźnie (rys.1.2.18).



Rys. 1.2.18. Wyznaczanie środka mas układu trzech punktów materialnych. Na rysunku jest to punkt C.

Każdy punkt materialny ma określoną masę  $m_i$  ( $m_1$ ,  $m_2$  i  $m_3$ ) oraz podane położenie  $(x_i, y_i)$ . Całkowita masa pokazanego układu jest równa:

$$M = m_1 + m_2 + m_3 = \sum m_i$$

Położenie środka masy C jest określone współrzędnymi wyliczonymi następująco:

$$x_s = \frac{m_1 x_1 + m_2 x_2 + m_3 x_3}{m_1 + m_2 + m_3}$$

$$y_s = \frac{m_1 y_1 + m_2 y_2 + m_3 y_3}{m_1 + m_2 + m_3}$$

Jak wynika z rysunku, współrzędne środka mas wyliczane są w konkretnym układzie współrzędnych, musimy mieć określony punkt odniesienia. W ogólnym przypadku  $n$  ciał środek masy  $n$  punktów materialnych leżących na płaszczyźnie wyznaczamy następująco:

$$x_s = \frac{\sum m_i x_i}{\sum m_i} = \frac{1}{M} \sum m_i x_i$$

$$y_s = \frac{\sum m_i y_i}{\sum m_i} = \frac{1}{M} \sum m_i y_i$$

W przestrzeni 3D środek mas wyznaczają trzy współrzędne  $x_s$ ,  $y_s$  i  $z_s$ , przy czym trzecia współrzędna wyliczana jest analogicznie;

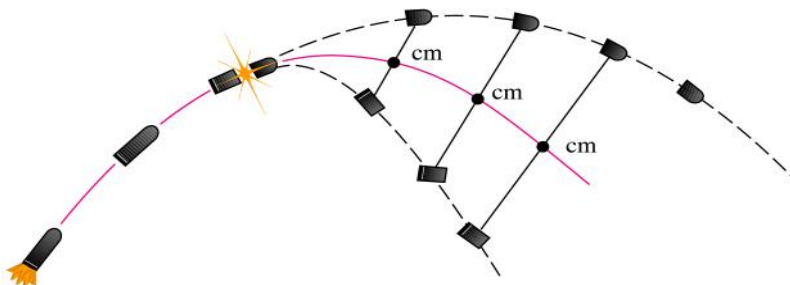
$$z_s = \frac{\sum m_i z_i}{\sum m_i} = \frac{1}{M} \sum m_i z_i$$

W praktyce, zamiast masy używamy ciężaru ciał - wyznaczamy środek ciężkości. Praktycznie oznacza to zastąpienie w powyższych wzorach masy ciężarem:

$$P = mg$$

gdzie  $g$  jest przyspieszeniem ziemskim na poziomie morza,  $g = 9.81 \text{ m/s}^2$ . Koncepcja środka masy jest wygodnym pojęciem szeroko wykorzystywanym w modelowaniu ruchu obiektów. Klasycznym przykładem jest modelowanie w grach komputerowych ruchu samochodu. Trajektoria ruchu samochodu jest obliczana właśnie dla środka masy. Można stosunkowo prosto wyliczyć położenie środka masy w samochodzie (oczywiście w wystarczającym przybliżeniu). Musimy znać rozmiar i wagę karoserii, rozmiar i wagę zbiornika na benzynę, rozmiar i wagę silnika, rozmiar i wagę kierowcy. Dla tych czterech elementów wyznaczamy położenia lokalnych środków mas, a na końcu obliczamy środek masy dla całego zestawu.

Modelując pokaz sztucznych ogni również możemy posłużyć się koncepcją ruchu środka mas (rys. 1.2.19). Rakieta startuje z ziemi i porusza się torem parabolicznym (ruch ciała w polu grawitacyjnym). Na pewnej wysokości następuje eksplozja i elementy rakiety zaczynają się poruszać w różnych kierunkach. W tym beładnym ruchu jest jednak porządek – zgodnie z zasadami fizyki, tory poszczególnych fragmentów muszą być takie, że ich środek masy porusza się dalej po torze parabolicznym, po jakim poruszałyby się rakieta, gdyby nie wybuchła.



Rys. 1.2.19. Rakieta eksploduje w locie. Środek masy elementów rakiety porusza się torem, po jakim poruszałyby się rakieta bez eksplozji.

Ogólny ruch układu punktów materialnych dzielimy na ruch postępowy jego środka masy i ruch obrotowy wokół środka masy. Ruch obrotowy wystąpi, gdy na przykład samochód wpadnie w poślizg i zacznie obracać się wokół własnej osi. W celu poprawnego modelowania ruchu obiektu, gdy występuje ruch obrotowy musimy uwzględnić dodatkowe zjawiska. Opis ruchu obrotowego jest bardziej skomplikowany niż opis ruchu postępowego. W dynamice znaczną rolę odgrywa pojęcie bezwładności (inercji), jest to własność ciał materialnych, dość istotna w pierwszej i drugiej zasadzie dynamiki Newtona. Jeżeli na ciało działa nie zrównoważony układ sił, to bezwładność przejawia się w tym, że zmiana ruchu nie zachodzi nagle, ale stopniowo, im większa jest bezwładność ciała, tym trudniej jest to ciało przyspieszyć. W ruchu postępowym masa jest miarą bezwładności. W opisie ruchu obrotowego posługujemy się pojęciem **momentu bezwładności**. Jest to wielkość charakteryzująca rozkład masy w ciele. Moment bezwładności jest miarą bezwładności ciała podczas ruchu obrotowego. Momentem bezwładności ciała względem wybranej osi (np. z) wyrażamy wzorem:

$$I_z = \sum m_i r_i^2$$

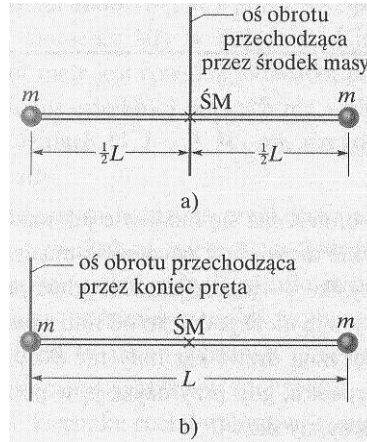
lub

$$I_z = \int \rho r^2 dV$$

gdzie  $m_i$  – masa punktu ciała,  $r$  lub  $r_i$  – odległość od osi obrotu,  $\rho$  - gęstość ciała,  $V$  – objętość ciała. Dla ciała rozciągniętego (np. szyba samochodu, krążek metalowy, itp.) moment bezwładności obliczyć można zastępując sumowanie – całkowaniem:

$$I_z = \int r^2 dm$$

Jak przykład obliczania momentu bezwładności rozważmy układ dwóch ciał, każde ma masę  $m$  i połączonych prętem o długości  $L$  i pomijalnej masie (rys. 1.2.20).



Rys. 1.2.20. Dwa ciała połączone prętem o znikomej masie, a) oś obrotu przechodzi przez środek mas, b) oś obrotu przechodzi przez środek ciała.

Najpierw obliczymy moment bezwładności w przypadku, gdy oś obrotu przechodzi przez środek mas. Odległość mas od osi obrotu jest równa  $0.5L$ , wobec tego moment bezwładności wynosi:


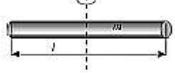




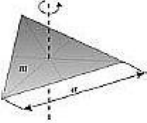

$$I = \sum m_i r_i^2 = m (0.5L)^2 + m (0.5L)^2 = \frac{1}{2} mL^2$$

W drugim przypadku, gdy oś obrotu przechodzi przez ciało umieszczone z lewej strony, moment bezwładności wynosi:

$$I = \sum m_i r_i^2 = m (0)^2 + mL^2 = mL^2$$

W zależności od wyboru osi, wokół której obraca się ciało, otrzymujemy różne wartości momentu bezwładności. W ogólności wyliczenie momentu bezwładności jest zadaniem skomplikowanym. Dla symetrycznych obiektów możemy skorzystać z odpowiednich tablic. W symulacji komputerowych, gdy posługujemy się modelami dość przybliżonymi, korzystanie z tablic jest uzasadnione. Okazuje się jednak, że momenty bezwładności wielu brył można obliczyć nie używając całek. W grafice komputerowej, bardzo często skomplikowane kształty obiektów, których ruch chcemy modelować, przybliżamy bardziej symetrycznymi kształtami, dla których dysponujemy tablicami momentów bezwładności.

Tabela 1. Momenty bezwładności wybranych ciał.

Pręt  $I = \frac{1}{3} \cdot m \cdot l^2$	Pręt  $I = \frac{1}{12} \cdot m \cdot l^2$
Walec  $I = \frac{1}{2} \cdot m \cdot r^2$	Krążek  $I = \frac{1}{4} \cdot m \cdot r^2$
Powłoka kulista  $I = \frac{2}{3} \cdot m \cdot r^2$	Stożek  $I = \frac{3}{10} \cdot m \cdot r^2$
Trójkąt  $I = \frac{1}{12} \cdot m \cdot a^2$	Sześcian  $I = \frac{1}{6} \cdot m \cdot a^2$

Wyliczenia momentów bezwładności można znacznie uprościć korzystając z następujących twierdzeń:

- moment bezwładności bryły jest wielkością addytywną (tzn. moment bezwładności bryły jest sumą momentów bezwładności części, na które daną bryłę można rozłożyć);
- twierdzenia Steinera: jeśli moment bezwładności bryły o masie  $m$  względem osi przechodzącej przez środek masy ciała wynosi  $I_0$ , to moment bezwładności tej bryły względem osi równoległej do danej osi i odległej od niej o  $d$  jest równy

$$I = I_0 + md^2$$

Możemy zilustrować twierdzenie Steinera ponownie wyliczając moment bezwładności dla układu dwóch ciał połączonych nieważkim prętem (rys. 1.2.20). Moment bezwładności takiego układu względem osi przechodzącej przez jedno ciało wyliczony ze wzoru Steinera (znany moment bezwładności dla tego układu względem osi przechodzącej przez środek masy układu) ma postać:

$$I = I_0 + md^2 = \frac{1}{2} mL^2 + (2m) \left( \frac{1}{2} L \right)^2 = mL^2$$

Otrzymaliśmy wynik identyczny jak w poprzednim przykładzie.

Tabela 2. Momenty bezwładności wybranych obiektów.

Momenty bezwładności I niektórych ciał	I
Pierścień, r i R są promieniami wewnętrznym i zewnętrznym, I względem osi pierścienia	$m(r^2+R^2)/2$
Stożek, I względem osi stożka	$3mR^2/10$
Obręcz, I względem osi obręczy	$mR^2$
Płyta prostokątna o rozmiarach a i b, I względem osi prostopadłej do płyty i przechodzącej przez jej środek	$m(a^2+b^2)/12$
Obręcz, I względem dowolnej średnicy	$mR^2/2$
Obręcz, I względem dowolnej linii stycznej	$3mR^2/2$

Uwzględnienie ruchu obrotowego jest ważne z punktu widzenia realistycznych symulacji. Gdy na przykład ciało się toczy to wykonuje zarówno ruch postępowy, jak i obrotowy. Dlatego też toczenie możemy traktować, jako złożenie ruchu postępowego i obrotowego. Ciało, które porusza się ruchem postępowym zachowuje się inaczej niż ciało poruszające się złożonym ruchem postępowo-obrotowym. Ilustruje to zagadnienie kolejny przykład.

Rozważmy krążek i kulę o masach m i promieniach R staczających się po równi pochyłej o wysokości h. Należy obliczyć ich prędkości u dołu równi.

Rozwiązując to zadanie wykorzystamy zasadę zachowania energii. Musimy uwzględnić zarówno energią kinetyczną ruchu postępowego jak i obrotowego.

Równanie zachowania energii ma postać:

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\omega^2$$

gdzie: g – przyspieszenie ziemskie, m- masa ciała, h - wysokość, v – prędkość w ruchu postępowym, I - moment bezwładności ciała,  $\omega$  - prędkość kątowa w ruchu po okręgu ( $d\theta/dt$ ).

Jeżeli skorzystamy z równości:

$$\omega = \frac{v}{R}$$

to otrzymamy:

$$mgh = \frac{1}{2}mv^2 + \frac{1}{2}I\left(\frac{v}{R}\right)^2$$



Z tego równania wyznaczamy prędkość:

$$v^2 = \frac{2mgh}{m + \frac{I}{R^2}}$$

Otrzymaliśmy ogólny wzór na prędkość ciała staczającego się po równi pochyłej. W celu otrzymania żądanej wartości prędkości musimy dysponować momentem bezwładności. Dla kuli  $I = 2mR^2/5$ , wobec czego:

$$v = \sqrt{\frac{10}{7}gh}$$

Dla krążka  $I = mR^2/2$ , wobec czego:

$$v = \sqrt{\frac{4}{3}gh}$$

Zauważmy, że w naszym przykładzie prędkość końcowa nie zależy od masy, zależy od momentu bezwładności, czyli kształtu obiektu. Gdyby ciała zsuwałyby się po równi pochyłej bez obrotów to prędkość końcowa byłaby jednakowa:

$$v = \sqrt{2gh}$$

Otrzymaliśmy trochę zaskakujący wynik – kula zsuwająca się po równi pochyłej ma na końcu większą prędkość niż kula tocząca się po równi.



---

# ROZDZIAŁ 2

## RACHUNEK RÓŻNICZKOWY I CAŁKOWY

---

2.1. Modele algebraiczne zjawisk fizycznych.....	36
2.2. Pochodne i różniczki .....	38
2.3. Rachunek całkowy .....	42
2.4. Rachunek różnicowy .....	47

---

## 2.1. Modele algebraiczne zjawisk fizycznych

Opisując świat realny, ustalamy związki zachodzące między wielkościami, które mogą przyjmować różne wartości. Dzięki analizie tych zależności i związków możemy rozwiązywać interesujące nas zadania.

Rozważmy ruch spadochroniarza. Spadochroniarz po wyskoczeniu z samolotu, do momentu otwarcia spadochronu spada swobodnie (jego prędkość wzrasta), po otwarciu spadochronu, prędkość jest znacznie mniejsza. Możemy zarejestrować czas (sekundy) i przebytą drogę (metry). Mając takie obserwacje możemy postawić pytanie o czas otwarcia spadochronu. Obserwacje przedstawiamy w postaci tabeli. W tabeli 2.1. pokazano obserwacje a także obliczoną drogę modelując zjawisko.

Tabela 2.1. Relacje pomiędzy czasem i przebytą drogą spadochroniarza.

Czas $t$	0	1	2	3	4	5
Przebyta droga $d$	0	4	16	36	46	52
Obliczona droga $d$	0	4	16	36	64	100

Otrzymaliśmy informacje w postaci zbioru par liczb. Jeżeli występuje jakiś związek między tymi parami, to opisaną sytuację możemy przedstawić w postaci **równania algebraicznego**. W naszym przypadku zauważamy, że w początkowej fazie spadku związek pomiędzy czasem i drogą przedstawia równanie:

$$d = (2t)^2 = 4t^2$$

Oznacza to, że ustaliliśmy zależność przebytej drogi od czasu a nasze równanie jest **modelem algebraicznym** ruchu spadochroniarza. Ta formuła obowiązuje do momentu otwarcia spadochronu. Zauważamy, że w czwartej sekundzie spadochroniarz przebył znacznie mniejszą drogę, niż to wynika ze wzoru. Było to spowodowane otwarciem spadochronu. Wobec tego sadzimy, że otwarto spadochron po 3 sekundach.

W przykładzie rozważaliśmy związek pomiędzy czasem i drogą spadochroniarza. Te wszystkie związki są funkcjami. Dla każdej chwili potrafimy podać drogę. Każdemu elementowi pierwszego zbioru (sekundy) odpowiada dokładnie jeden element drugiego zbioru (droga). Pierwszy zbiór nazywa się **dziedziną**, a drugi **przeciwdziedziną**. Często używamy notacji strzałkowej:

$$t \rightarrow 4t^2$$

Dziedzina funkcji musi być wyraźnie określona. Elementy przeciwdziedziny tworzą **zbiór wartości** tej funkcji.

Dla funkcji:

$$t \rightarrow 4t^2$$

dziedzina ma postać:

$$\{t: 0 \leq t \leq 3\}$$

a zbiorem wartości jest

$$\{d: 0 \leq d \leq 36\}$$

Zagadnienie spadku swobodnego możemy zanalizować bardziej ogólnie. Zakładamy, że spadek swobodny ciężkiego punktu materialnego występuje w próżni. Czas  $t$  (w sekundach) liczymy od początku spadania, drogę  $s$ , przebytą przez punkt wyrażamy wzorem:

$$s = \frac{1}{2}gt^2$$

gdzie  $g = 9.81 \text{ m/s}^2$  oznacza przyspieszenie ziemskie.

Oznaczając przez  $\Delta t$  przyrost czasu a przez  $\Delta s$  przyrost drogi, mamy wyrażenie:

$$s + \Delta s = \frac{1}{2}g(t + \Delta t)^2$$

Wykonując przekształcenia otrzymujemy wyrażenie:

$$\Delta s = gt\Delta t + \frac{1}{2}g\Delta t^2$$

Dzieląc przyrost drogi przez przyrost czasu otrzymamy prędkość średnią na odcinku  $\Delta s$ :

$$v_{\text{sr}} = \frac{\Delta s}{\Delta t} = gt + \frac{1}{2}g\Delta t$$

Prędkość średnia zależy od  $\Delta t$ , im mniejszy jest przyrost czasu, tym dokładniejsze jest oszacowanie prędkości średniej.

Prędkością  $v$  punktu w chwili  $t$  nazywamy granicę, do której dąży prędkość średnia, gdy  $\Delta t$  dąży do zera:

$$v = \lim_{\Delta t \rightarrow 0} \left( gt + \frac{1}{2}g\Delta t \right) = gt$$

W ten sposób można obliczać prędkość punktu w ruchu.

Położenie punktu wyznaczone jest odległością  $s$ , którą mierzymy od punktu początkowego ruchu. Czas  $t$  liczymy od ustalonej chwili początkowej.

Ruch jest w zupełności opisany, jeżeli znamy równanie ruchu:

$$s = f(t)$$

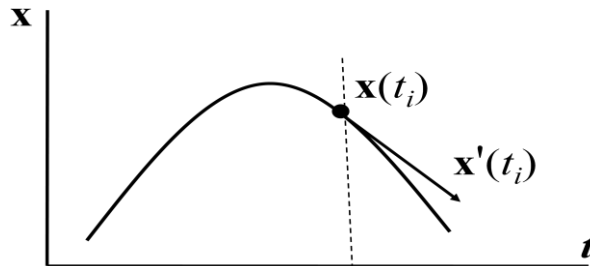
Do wyznaczenia prędkości punktu w danej chwili  $t$  posługujemy się stosunkiem przyrostu drogi do przyrostu czasu:

$$\frac{\Delta s}{\Delta t}$$

W ten sposób określamy *prędkość średnią*. Przechodząc do granicy, otrzymujemy *prędkość rzeczywistą*:

$$v = \lim_{\Delta t \rightarrow 0} v_{sr} = \lim_{\Delta t \rightarrow 0} \frac{\Delta s}{\Delta t}$$

Jeżeli znamy funkcję położenia obiektu  $\mathbf{x}(t)$ , możemy posłużyć się pojęciem pochodnej. Pochodna oznaczona symbolem  $\mathbf{x}'(t)$  opisuje jak  $\mathbf{x}$  zmienia się, gdy zmienia się czas  $t$  (mamy także symbol  $d\mathbf{x}/dt$ ). Na wykresie zależności położenia od czasu pochodna  $\mathbf{x}'(t)$  określa wektor styczny do krzywej w czasie  $t$  (rys. 2.1.)



Rys.2.1 Funkcja położenia i jej pochodna w punkcie

## 2.2. Pochodne i różniczki

W prostych zagadnieniach modelowania i symulacji, stosowanych w grafice komputerowej i grach komputerowych znajomość rachunku różniczkowego i całkowego nie jest wymagana. Wystarczają podstawowe pojęcia mechaniki, takie jak droga, prędkość, czy przyspieszenie. W zastosowaniach praktycznych posługujemy się prostymi równaniami, np. z zestawem równań kinematycznych dla przypadku przyspieszenia stałego.

W tworzeniu realistycznych gier komputerowych, gdy należy opracowywać odpowiednie algorytmu dla zjawisk dynamicznych, musimy posiadać, przynajmniej na poziomie elementarnym, znajomość zasad rachunku różniczkowego i całkowego.

Kluczowym pojęciem rachunku różniczkowego jest pojęcie pochodnej.

Pochodna funkcji

$$y = f(x)$$

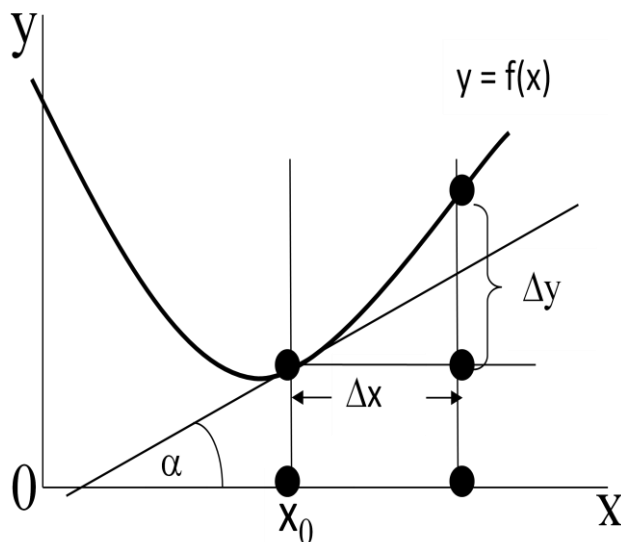
w punkcie  $x_0$  jest granicą, do której dąży iloraz różnicowy

$$\frac{\Delta y}{\Delta x} = \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

czyli

$$\lim_{\Delta x \rightarrow 0} \frac{\Delta y}{\Delta x} = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}$$

Pochodna ma prostą interpretację geometryczną. Pochodna funkcji  $f(x)$  w punkcie  $x_0$  jest równa tangensowi kąta  $\alpha$ , który tworzy styczna do wykresu funkcji  $f$  w punkcie  $x_0$  z osią  $OX$  (rys. 2.2.).



Rys.2.2. Interpretacja geometryczna pochodnej.

W podręcznikach występują różne symbole na oznaczenie pochodnej:

	symbol	
G.W. Leibniz	$\frac{dy}{dx}$	$\frac{df(x_0)}{dx}$
J.L. Lagrange	$y'$	$f'(x_0)$
A. Cauchy	$Dy$	$Df(x_0)$

Pochodne prostych funkcji są znane:

Funkcja	pochodna
$y = x$	$y' = 1$
$y = c$	$y' = 0$
$y = x^a$	$y' = ax^{a-1}$
$y = \frac{1}{x}$	$y' = -\frac{1}{x^2}$
$y = e^x$	$y' = e^x$
$y = \ln x$	$y' = \frac{1}{x}$
$y = \sin x$	$y' = \cos x$
$y = \cos x$	$y' = -\sin x$

W modelowaniu zjawisk fizycznych ważną rolę odgrywają różniczki (nie należy ich mylić z pochodnymi). Rozważmy funkcję  $y = f(x)$ , określoną w pewnym przedziale i ciągłą w rozpatrywanym punkcie  $x_0$ . Przyrostowi  $\Delta x$  argumentu odpowiadać będzie przyrost

$$\Delta y = \Delta f(x_0) = f(x_0 + \Delta x) - f(x_0)$$

nieskończenie mały z  $\Delta x$ . Zgodnie z G.M. Fichtenholzem pytamy się czy istnieje dla  $\Delta y$  wielkość nieskończenie mała  $A\Delta x$  ( $A = \text{const}$ ) liniowa względem  $\Delta x$  i taka, że różnica  $\Delta y - A\Delta x$  jest nieskończenie małą rzędu wyższego niż  $\Delta x$ , tzn. że

$$\Delta y = A\Delta x + o(\Delta x)$$

Jeżeli dla  $A \neq 0$ , zachodzi ta równość, to mówimy, że funkcja  $y=f(x)$  jest różniczkowalna, wyrażenie  $A\Delta x$  nazywa się **różniczką funkcji** i oznacza się ją symbolem  $dy$  lub  $df(x_0)$ .

Obliczanie różniczki zilustrujemy dwoma przykładami.



Przykład 1.

Wzór na pole koła  $P$  o promieniu  $r$  ma postać:  $P = \pi r^2$ . Jeżeli zwiększymy promień, zwiększymy pole koła. Niech zmiana promienia wynosi  $\Delta r$ , co odpowiada przyrostowi pola  $\Delta P$ . Obliczamy ten przyrost:

$$\Delta P = \pi(r + \Delta r)^2 - \pi r^2 = 2\pi r \Delta r + \pi(\Delta r)^2$$

Część główna przyrostu pola  $\Delta P$  wynosi  $2\pi r \Delta r$ , gdy zmiana promienia  $\Delta r$  dąży do zera. Oznacza to, że poszukiwana różniczka wyraża się wzorem:

$$dP = 2\pi r \Delta r$$

Mamy, więc następujące wzory:

$$\text{Pole koła: } P = 2\pi r^2$$

$$\text{Pochodna: } P'_r = 2\pi r$$

$$\text{Różniczka: } dP = 2\pi r \Delta r$$

Przykład 2.

W spadku swobodnym wzór na drogę  $s$  ma postać ( $t$  – czas,  $g$  – przyspieszenie):

$$s = \frac{1}{2}gt^2$$

Przyrost drogi podczas spadku swobodnego w czasie  $\Delta t$  wynosi (obliczenia wykonujemy dla przedziału czasowego  $(t, t+\Delta t)$ ):

$$\Delta s = \frac{1}{2}g(t + \Delta t)^2 - \frac{1}{2}gt^2 = gt\Delta t + \frac{1}{2}g(\Delta t)^2$$

Część główna przyrostu drogi  $\Delta s$  wynosi  $gt\Delta t$ , gdy przyrost czasu  $\Delta t$  dąży do zera (przypominamy także, że  $v = gt$ , tzn. różniczka drogi jest iloczynem prędkości  $v$  i przyrostu czasu  $\Delta t$ ).

Mamy więc następujące wzory:

$$\text{droga: } s = \frac{1}{2}gt^2$$

$$\text{Pochodna: } s'_t = gt$$

$$\text{Różniczka: } ds = gt\Delta t$$

### 2.3. Rachunek całkowy

Całka jest jednym z kluczowych pojęć matematyki. Pojęcie całki zostało wprowadzone przez G.W. Leibniza i I. Newtona w XVII wieku. Precyzyjnie całka została zdefiniowana w XIX wieku (między innymi całka oznaczona Lebesgue'a).

Funkcja  $F(x)$  nazywa się **funkcją pierwotną** funkcji  $f(x)$  lub **całką** w danym przedziale, jeśli w całym tym przedziale  $f(x)$  jest pochodną funkcji  $F(x)$  lub, co na jedno wychodzi,  $f(x)dx$  jest różniczką  $F(x)$ :

$$F'(x) = f(x) \quad \text{lub} \quad dF(x) = f(x)dx$$

Znalezienie wszystkich funkcji pierwotnych danej funkcji nazywa się **całkowaniem**. Jest to zagadnienie odwrotne do zagadnienia rachunku różniczkowego. Całkę nieoznaczoną funkcji  $f(x)$  oznaczamy symbolem:

$$\int f(x)dx$$

Całką nieoznaczoną nazywa się zbiór funkcji  $F(x)$  (tzw. funkcji pierwotnych) spełniających równanie:

$$[F(x)]' = f(x)$$

(pochodna funkcji  $F(x)$  równa się funkcji podcałkowej  $f(x)$ ).

Dowolne funkcje pierwotne danej funkcji  $f(x)$  różnią się stałą liczbą, zatem formalnie musimy napisać:

$$\int f(x)dx = F(x) + C$$

( $C$  jest to stała całkowania)

Wyliczanie całek nieoznaczonych prostych funkcji nie nastręcza trudności. Dla przykładu, gdy mamy funkcję postaci  $f(x) = x^2$ , to jej całka nieoznaczona ma postać:

$$\int x^2 dx = \frac{x^3}{3} + C$$

co można łatwo sprawdzić wykonując różniczkowanie.

Całki nieoznaczone prostych funkcji są znane:

$$\begin{aligned}\int 0 \cdot dx &= C \\ \int dx &= x + C \\ \int x^a dx &= \frac{x^{a+1}}{a+1} + C \\ \int \frac{1}{x} dx &= \ln|x| + C \\ \int \sin x dx &= -\cos x + C \\ \int \frac{1}{\cos^2 x} dx &= \tan x + C\end{aligned}$$

Całki i różniczki są eleganckimi narzędziami do opisu zjawisk fizycznych. Te narzędzia są szeroko stosowane w mechanice. Rozpatrzmy przypadek, gdy znane jest równanie ruchu  $s = s(t)$ , gdzie  $s$  jest drogą a  $t$  czasem. W takim przypadku możemy przy pomocy różniczkowania określić prędkość, jako  $v = ds/dt$  oraz przyspieszenie, jako  $a = dv/dt$ . Często mamy zadanie odwrotne. Znamy przyspieszenie  $a$  jako funkcję czasu, tzn.  $a = a(t)$ , a chcemy wyznaczyć prędkość  $v$  oraz drogę  $s$  w zależności od czasu. Wobec tego mamy:

$$\begin{aligned}v &= \int a(t) dt \\ s &= \int v(t) dt\end{aligned}$$

Jeżeli rozważamy spadek swobody pod wpływem siły ciężkości to  $a = g$  i wtedy:

$$v = \int g dt = gt + C$$

Jeżeli wiemy, że w chwili  $t=t_0$  prędkość  $v = v_0$ , to mamy:

$$v_0 = gt_0 + C$$

i możemy wyliczyć stałą  $C$ :

$$C = v_0 - gt_0$$

Po wyznaczeniu stałej  $C$  ostatecznie mamy określoną prędkość:

$$v = g(t - t_0) + v_0$$

Możemy ustalić wzór na drogę  $s$ :

$$s = \int [g(t - t_0) + v_0] dt = \frac{1}{2} g(t - t_0)^2 + v_0(t - t_0) + C'$$

Stałą wyznaczamy z warunku, że dana jest drogą  $s=s_0$  w chwili  $t = t_0$ . Wtedy  $C=s_0$ :

$$s = \frac{1}{2}g(t - t_0)^2 + v_0(t - t_0) + s_0$$

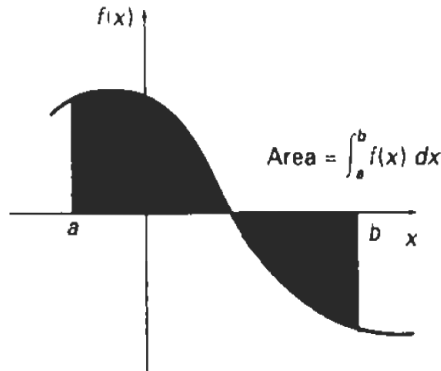
wartości  $t_0$ ,  $s_0$ ,  $v_0$  nazywają się wartościami początkowymi. Dla zerowych wartości początkowych mamy wzór:

$$s = \frac{1}{2}gt^2$$

W modelowaniu zjawisk fizycznych ważne są **całki oznaczone**. Całką oznaczoną funkcji  $f(x)$  w granicach od  $a$  do  $b$  nazywa się różnicę  $F(b) - F(a)$ , gdzie  $F(x)$  jest dowolną funkcją pierwotną dla  $f(x)$ .

Dla ciągłej funkcji  $f(x)$ , całka oznaczana ma prostą interpretację geometryczną – jest to pole obszaru między krzywą  $y = f(x)$ , osią  $OX$  i prostopadłymi do niej wystawionymi w punktach  $a$  i  $b$ . Polu położonemu nad osią  $OX$  przypisujemy znak  $+$ , polu poniżej tej osi przypisujemy znak  $-$ .

Interpretacja geometryczna całki oznaczonej pokazana jest na rysunku 2.3.



Rys. 2.3. Interpretacja geometryczna całki oznaczonej

Nie wszystkie całki funkcji ciągłych dają się prosto obliczyć. W takim przypadku musimy korzystać z metod przybliżonych.

Całka z  $f(x)$  może być numerycznie przybliżona wzorem sumacyjnym:

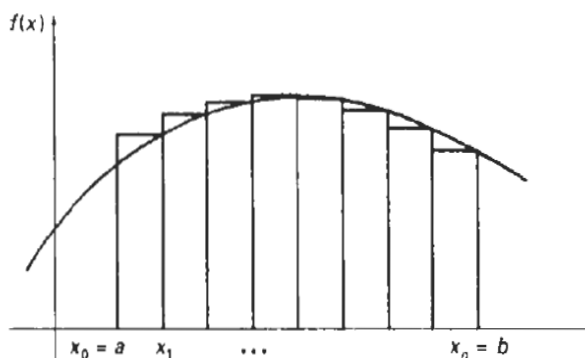
$$\int_b^a f(x)dx = \sum_{k=1}^n f_k(x)\Delta x_k$$

W powyższym wzorze  $f_k(x)$  jest przybliżeniem  $f(x)$  na przedziale  $\Delta x_k$ .

Do szacowania numerycznego całek oznaczonych można stosować wiele metod:

- Metoda prostokątów. Możemy w każdym przedziale przybliżać funkcję  $f(x)$  stałą wartością, i sumować wszystkie pola powierzchni prostokątów. Im więcej będzie prostokątów (mniejszy krok) tym dokładniejsze będzie oszacowanie.
- Metoda trapezów. Jeżeli pole powierzchni będziemy przybliżali sumą trapezów, otrzymamy metodę trapezów, która generalnie daje mniejszy błąd niż metoda prostokątów.
- Metoda Simpsona. Jeżeli użyjemy paraboli do przybliżenia przebiegu funkcji w każdym przedziale, to otrzymamy metodę Simpsona.

Ilustracja metody prostokątów pokazana jest na rysunku 2.4. W tej metodzie obszar ograniczony funkcją dzieli się na prostokąty, wylicza się pola poszczególnych prostokątów i sumuje. Daje to szybka metodę szacowania całek oznaczonych.

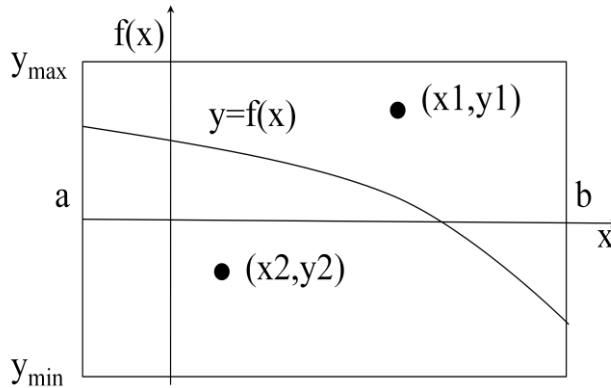


Rys. 2.4. Metoda prostokątów całkowania numerycznego

W metodzie Simpsona całka z  $f(x)$  może być numerycznie przybliżona wzorem sumacyjnym:

$$\int_b^a f(x) dx \approx \frac{\Delta x}{3} \left[ f(a) + f(b) + 4 \sum_{\text{odd } k=1}^{n-1} f(x_k) + 2 \sum_{\text{even } k=2}^{n-2} f(x_k) \right]$$

Elegancką metodą szacowania całek oznaczonych jest metoda Monte Carlo. W metodzie Monte-Carlo możemy całkowaną funkcję  $f(x)$  zamknąć w prostokącie na przedziale od  $a$  do  $b$  i losować punkty (wykorzystując generatory liczb pseudolosowych).



Rys. 2.5 Ilustracja metody Monte Carlo

W metodzie Monte Carlo generujemy (losujemy)  $n$  punktów. Punkt albo jest w polu powyżej funkcji, albo poniżej. Jeżeli oznaczymy wszystkie wygenerowane punkty przez  $n$  a przez  $n_{\text{count}}$  punkty, które leżą poniżej funkcji  $f(x)$ , to oszacowanie całki ma postać:

$$\int_b^a f(x) dx \approx h(b-a) \frac{n_{\text{count}}}{n}$$

Przypadkowe położenie punktu  $(x,y)$  w obszarze prostokąta jest najpierw generowane przy pomocy liczb losowych  $r_1$  i  $r_2$ . Kolejno wyliczamy:

$$\begin{aligned} h &= y_{\text{max}} - y_{\text{min}} \\ x &= a + r_1(b-a) \\ y &= y_{\text{min}} + r_2 h \end{aligned}$$

Błąd oszacowania maleje z ilością wygenerowanych punktów. Opisana metoda Monte –Carlo jest doskonałą metodą szacowania całek wielokrotnych.

W tej metodzie bardzo ważny jest tzw. generator liczb pseudolosowych. Możemy korzystać z doskonałych algorytmów generowania liczb losowych publikowanych w literaturze przedmiotu, możemy korzystać z funkcji bibliotecznych używanych kompilatorów, możemy też sami napisać prosty generator. Jednym z najprostszych jest **linear congruential generator**:

$$i_k = ai_{k-1} + c(\text{mod } m)$$

W tym wzorze parametry  $a$ ,  $c$ ,  $m$ , oraz  $i_0$  są liczbami całkowitymi, a  $i_0$  jest wartością startową (ang. *seed*). Parametr  $m$  wybieramy tak, aby był jak największy w konkretnym komputerze. Także wartości dla  $a$  oraz  $c$  wybieramy, tak, aby okres generatora był jak największy. Na przykład dla maszyny z 32-bitową liczbą całkowitą, wykorzystując wieloletnie doświadczenie możemy wybrać:  $m = 232$ ,  $a = 1664525$ , oraz  $c = 1013904223$ .

## 2.4. Rachunek różnicowy

Gdy komputery i metody numeryczne zyskały pełnie praw, został znacząco rozwinięty rachunek **różnicowy**, będący odpowiednikiem tradycyjnego **rachunku różniczkowego**.

Wprowadzimy elementy rachunku różnicowego zgodnie z podejściem demonstrowanym w podręczniku „Matematyka konkretna”, PWN, Warszawa, 1996.

Należy pamiętać, że sukcesy analizy matematycznej zasadniczo oparte są na dwóch faktach: mamy pojęcie **funkcji ciągłych** oraz pojęcie **nieskończoności**.

W systemach komputerowych operujemy **dyskretnym** zbiorem liczb oraz każdy taki zbiór jest **skończony**. W zastosowaniach praktycznych z wykorzystaniem komputerów decydujące znaczenie mają **metody numeryczne**.

W rachunku różniczkowym najważniejszym pojęciem jest operator pochodnej  $D$  (zapis Cauchy’ego). Jego postać to:

$$Df(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

W rachunku różnicowym najważniejszym pojęciem jest operator różnicy  $\Delta$ . Jego postać to:

$$\Delta f(x) = f(x+1) - f(x)$$

Operatory  $D$  i  $\Delta$  są funkcjami, które produkują funkcje. Gdy  $f(x)$  jest funkcją gładką ze zbioru liczb rzeczywistych, to  $Df(x)$  też jest funkcją w tym zbiorze. Tak samo, jeżeli  $f(x)$  jest jakąkolwiek funkcją w zbiorze liczb rzeczywistych, to  $\Delta f(x)$  też będzie taką funkcją. Wartości  $Df$  i  $\Delta f$  w punkcie  $x$  są, więc określone zgodnie z podanymi wzorami. W rachunku różniczkowym operator  $D$  ma operator działający odwrotnie – całkę (symbol  $\int$ ). Związek pomiędzy całką a pochodną ma postać:

$$g(x) = Df(x) \quad \Leftrightarrow \quad \int g(x)dx = f(x) + C$$

Dla operatora różnicowego  $\Delta$  istnieje także operator odwrotny, antyróżnica (albo suma)  $\Sigma$ , ich relacje są następujące:

$$g(x) = \Delta f(x) \quad \Leftrightarrow \quad \Sigma g(x)\delta x = f(x) + C$$

Rolę sumy nieoznaczonej dla funkcji  $g(x)$  pełni  $\Sigma(\dots)$ , będąca klasą funkcji, dla których różnica wynosi  $g(x)$ . Symbol  $\delta$  dla operatora  $\Delta$  pełni taką samą rolę jak symbol  $d$  dla operatora  $D$ . Symbol  $C$  w przypadku sum nieoznaczonych oznacza dowolną funkcję okresową  $p(x)$  taką, że  $p(x+1) = p(x)$ . Przykład takiej funkcji to np.:

$$a + b \sin 2\pi x$$

W rachunku różniczkowym, jeśli  $g(x)=Df(x)$  to mamy całkę oznaczoną:

$$\int_a^b g(x) dx = \left| \right|_a^b = f(b) - f(a)$$

W rachunku różnicowym istnieje także odpowiednik całki oznaczonej – **suma oznaczona**, jeśli  $g(x)=\Delta f(x)$  to

$$\sum_a^b g(x)\delta x = f(x) \Big|_a^b = f(b) - f(a)$$

Korzystając z podejścia prezentowanego w podręczniku „Matematyka konkretna” (R.Graham, D. Knuth, O. Patashnik, PWN, 1996) omówimy w skrócie własności sumy oznaczonej. Niech (jest to definicja operatora różnicowego):

$$g(x) = \Delta f(x) = f(x + 1) - f(x)$$

Jeżeli  $b = a$  to

$$\sum_a^a g(x)\delta x = f(a) - f(a) = 0$$

Jeżeli  $b = a+1$  to:

$$\sum_a^{a+1} g(x)\delta x = f(a + 1) - f(a) = g(a)$$

Ogólnie, jeżeli  $b$  zwiększy się o 1, to

$$\sum_a^{b+1} g(x)\delta x - \sum_a^b g(x)\delta x = (f(b + 1) - f(a)) - (f(b) - f(a))$$

$$(f(b + 1) - f(a)) - (f(b) - f(a)) = f(b + 1) - f(b) = g(b)$$

Dzięki temu przykładowi, wiemy co dokładnie reprezentuje **suma oznaczona** dla całkowitych  $a$  i  $b$ , spełniających warunek  $b \geq a$  :

$$\sum_a^b g(x)\delta x = \sum_{k=a}^{b-1} g(k) = \sum_{a \leq k < b} g(k), \quad \text{całkowite } b \geq a$$

Suma oznaczona oznacza to samo, co zwykła suma, tyle, że pomijamy składnik odpowiadający górnemu ograniczeniu.



---

# ROZDZIAŁ 3

## METODY NUMERYCZNE

---

3.1. Szereg Taylora .....	50
3.2. Proste metody całkowania numerycznego .....	51
3.3. Rozwiązywanie równań różniczkowych zwyczajnych metodami numerycznymi.....	62

---

### 3.1. Szereg Taylora

Szeregi Taylora stanowią fundament metod numerycznych. Wiele technik stosowanych w metodach numerycznych korzysta wprost z szeregów Taylora, na przykład do oszacowania błędów.

Jeżeli funkcja  $f(x)$  ma  $n$ -tą pochodną  $f^{(n)}(x)$  w pewnym domkniętym przedziale zawierającym punkt  $a$ , wówczas dla każdego  $x$  z tego przedziału mamy następujący wzór:

$$f(x) = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \dots \\ + \frac{f^{(n-1)}}{(n-1)!}(x-a)^{n-1} + \frac{f^{(n)}}{n!}(x-a)^n$$

Jest to **wzór Taylora**. Ostatni wyraz we wzorze Taylora jest nazywany **resztą wzoru Taylora**:

$$R_n = \frac{f^{(n)}(c_n)}{n!}(x-a)^n$$

Jeżeli w szeregu Taylora przyjmiemy  $a = 0$ , to otrzymamy **szereg Maclaurina**:

$$f(x) = f(0) + \frac{f'(0)}{1!}x + \frac{f''(0)}{2!}x^2 + \dots + \frac{f^{(n)}(0)}{(n)!}x^n$$

Wykorzystamy szereg Maclaurina aby rozwinąć funkcje  $\sin(x)$ . Zgodnie ze wzorem mamy:

$$\sin(x) = \sin(0) + x \cos(0) - \frac{x^2}{2!} \sin(0) - \frac{x^3}{3!} \cos(0) + \\ + \frac{x^4}{4!} \sin(0) + \frac{x^5}{5!} \cos(0) + \dots$$

Ponieważ  $\sin(0) = 0$  oraz  $\cos(0) = 1$ , ostatecznie mamy:

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} + \dots$$

Rozwinięcie funkcji  $e^x$  na szereg Maclaurina ma postać:

$$e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \dots$$

### 3.2. Proste metody całkowania numerycznego

W wielu przypadkach modelowania i symulacji zjawisk fizycznych nie potrafimy scałkować funkcji analitycznie. W takich przypadkach posługujemy się całkowaniem numerycznym. Metody numeryczne pozwalają oszacować całki funkcji z dość dużą dokładnością. Numeryczne całkowanie jest także podstawową metodą w przypadku, gdy chcemy oszacować całkę funkcji daną w postaci dyskretnych punktów. Ze względu na wagę problemu, istnieje wiele metod całkowania numerycznego, ale w naszym podręczniku opiszemy tylko te najprostsze.

#### Wzór prostokątów

Jest to bardzo prosta metoda całkowania numerycznego. Oszacowanie całki przy pomocy wzoru prostokątów ( $a$  i  $b$  są to granice całkowania,  $n$  – ilość przedziałów):

$$I = \int_a^b f(x) dx$$

ma postać:

$$I \approx h \sum_{i=0}^{n-1} f(x_i), \quad h = \frac{b-a}{n}$$

Ulepszona metoda punktów środkowych (tzw. *metoda midpoint*) ma postać:

$$I \approx h \sum_{i=1}^n f\left(x_i - \frac{h}{2}\right), \quad h = \frac{b-a}{n}, \quad x_i = x_0 + hi, \quad x_0 = a$$

Implementacja funkcji do szacowania całek za pomocą wzoru prostokątów typu *midpoint* prezentowana jest poniżej.

---

**Listing 3.1. Całkowanie numeryczne metodą midpoint**


---

```
// całkowanie numeryczne, metoda midpoint
//a, b – granice całkowania, n – liczba podprzedziałow

double prost (double (*f) (double), double a, double b, int n)

{ double s=0, h=(b-a)/(double)n, h2=h/2, xi;
  for (int i=1; i <=n; i++)
    { xi=a+i*h - h2;
      s+=f(xi);
    }
  return s*h;
}
```

---

Pierwszym argumentem funkcji **prost()** jest wskaźnik do funkcji podcałkowej, funkcja podcałkowa musi być dostarczona w postaci jawnej.

Zastosujemy funkcję z listingu 3.1. do oszacowania następującej całki

$$I \approx \int_0^1 x^3 dx$$

Prototyp funkcji **prost** wymaga wskaźnika do funkcji oraz liczby przedziałów **n**:

*double prost (double (\*f)(double), double a, double b, int n);*

Postać funkcji podcałkowej jest następująca:

```
double fx(double x)
{ return x*x*x;
}
```

Wywołanie funkcji ma postać:

*prost(fx,0,1,5)*

gdzie *fx* jest funkcją podcałkową, drugi argument jest dolną granicą całkowania (*a= 0*), trzeci argument jest górną granicą całkowania (*b=1*), czwarty argument jest liczbą przedziałów (*n= 5*).

Program pokazany na listingu 3.2 szacuje naszą całkę metodą prostokątów z opisanymi argumentami.

Listing 3.2. Całkowanie numeryczne, metoda midpoint, funkcja  $x^3$ 

```
//metoda prostokatow, midpoint

#include <iostream.h>
#include <conio.h>

double prost (double (*f)(double), double a, double b, int n);

double fx(double x) //funkcja podcalkowa
{return x*x*x ;
}

int main()
{ cout << "wartosc calki =" << prost(fx,0,1,5)<< endl; //0.25000
  getch();
  return 0;
}

double prost (double (*f) (double), double a, double b, int n)
{ double s=0, h=(b-a)/(double)n, h2=h/2, xi;
  for (int i=1; i <=n; i++)
    { xi=a+i*h - h2;
      s+=f(xi);
    }
  return s*h;
}
```

Po uruchomieniu programu mamy wynik:

wartość całki = 0.245

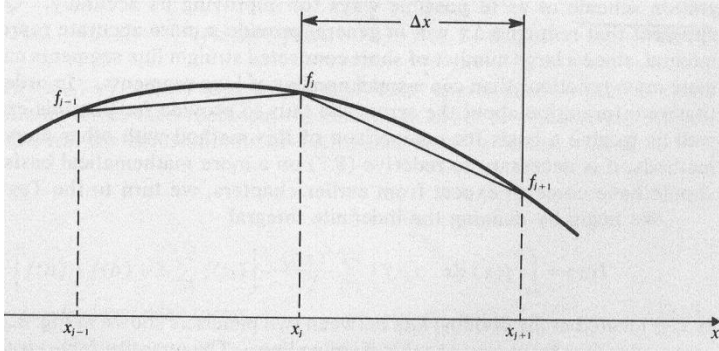
Znamy także, wartość całki obliczoną analitycznie:

wartość dokładna = 0.250

Oszacowanie numeryczne różni się od wartości prawdziwej o 0.005. Jeżeli zwiększymy liczbę przedziałów do 20, to oczywiście wynik się znacznie poprawi:

wartość obliczona,  $n = 20$ :  $I \approx 0.249688$

Kolejną prostą metodą całkowania numerycznego jest metoda trapezów. Pole pod całką jest sumą pól trapezów (rys. 3.1).



Rys.3.1. Całkowanie numeryczne - metoda trapezów

Jest to także bardzo prosta metoda całkowania numerycznego. Oszacowanie całki

$$I = \int_a^b f(x) dx$$

przy pomocy wzoru trapezów (a i b są to granice całkowania, n – ilość przedziałów) ma postać:

$$I \approx h \left( \frac{f(x_0)}{2} + \frac{f(x_n)}{2} + \sum_{i=1}^{n-1} f(x_i) \right)$$

$$h = \frac{b-a}{n}, \quad x_i = x_0 + hi, \quad x_0 = a$$

Nazwa metody związana jest z faktem, że sumujemy pola trapezów (pole  $s = 0.5 \cdot (a+b) \cdot h$ ):

$$T(h) = \sum_{i=1}^n \frac{f(x_{i-1}) + f(x_i)}{2} (x_i - x_{i-1})$$

Implementacja funkcji do szacowania całek za pomocą wzoru trapezów prezentowana jest poniżej.

---

**Listing 3.3. Całkowanie numeryczne metodą trapezów**

---

```
// calkowanie numeryczne, metoda trapezow
//a, b – granice calkowania, n – liczba podprzedzialow

double trapez (double (*f)(double), double a, double b, int n)
{double s=0.5*(f(a)+f(b));
 double h = (b-a)/(double)n;
 double xi;
 for (int i=1; i <=n-1; i++)
   {xi=a+i*h ;
    s+=f(xi);
   }
 return s*h;
}
```

---

Zastosujemy funkcję z listingu 3.3. do oszacowania znanej nam całki

$$I \approx \int_0^1 x^3 dx$$

Prototyp funkcji *trapez* wymaga wskaźnika do funkcji, zakresu całkowania oraz liczby przedziałów *n*:

```
double trapez (double (*f)(double), double a, double b, int n);
```

Postać funkcji podcałkowej jest następująca:

```
double fx(double x)
{ return x*x*x;
}
```

Wywołanie funkcji ma postać:

```
trapez(fx,0,1,5)
```

gdzie *fx* jest funkcją podcałkową, drugi argument jest dolną granicą całkowania (*a*= 0), trzeci argument jest górną granicą całkowania (*b*=1), czwarty argument jest liczbą przedziałów (*n*= 5).

Program pokazany na listingu 3.4 szacuje naszą całkę metodą trapezów z opisanymi argumentami.

Listing 3.4. Całkowanie numeryczne, metoda trapezów, funkcja  $x^3$ 

```
// metoda trapezow
#include <iostream.h>
#include <conio.h>

double fx(double x)
{return x*x*x;
}
double trapez (double (*f)(double), double a, double b, int n);

int main()
{ cout << "wartosc calki =" << trapez(fx,0,1,5)<< endl;
  getch();
  return 0;
}

double trapez (double (*f)(double), double a, double b, int n)
{double s=0.5*(f(a)+f(b));
double h = (b-a)/(double)n;
double xi;
for (int i=1; i <=n-1; i++)
  { xi=a+i*h;
    s+=f(xi);
  }
return s*h;
}
```

Po uruchomieniu programu mamy wynik:

wartość całki = 0.260

Znamy wartość całki obliczoną analitycznie:

wartość dokładna = 0.250

Oszacowanie numeryczne różni się od wartości prawdziwej o 0.01. Jeżeli zwiększymy liczbę przedziałów do 20, to wynik się znacznie poprawi:

wartość obliczona,  $n = 20$ :  $I \approx 0.2506625$

Kolejną stosunkowo prostą metodą szacowania całek jest metoda Simpsona. W odróżnieniu od metody trapezów, która funkcję w zadanym przedziale



aprosymuje prostą, w metodzie Simpsona wykorzystujemy łuk paraboli to tego celu.

Oszacowanie całki

$$I = \int_a^b f(x) dx$$

przy pomocy wzoru Simpsona (a i b są to granice całkowania, n – ilość przedziałów) jest dane wzorem:

$$I \approx \frac{h}{3} \left( f(a) + f(b) + 4 \sum_{i=1, \text{odd}}^{n-1} f[x_i] + 2 \sum_{i=2, \text{even}}^{n-2} f(x_i) \right)$$

$$h = \frac{b-a}{n}, \quad x_i = x_0 + hi, \quad x_0 = a$$

W tej metodzie liczba przedziałów musi być parzysta (ang. *odd* - nieparzysta, *even* - parzysta).

Implementacja funkcji do szacowania całek za pomocą wzoru Simpsona prezentowana jest poniżej.

#### Listing 3.5. Całkowanie numeryczne metodą Simpsona

---

```
// całkowanie numeryczne, metoda Simpsona
//a, b – granice całkowania, nn – liczba podprzedziałow

double simp (double (*f)(double), double a, double b, int nn)
{int n;
 double s1=0, s2=0, h;
 if (nn%2 != 0) {cout << "nieparzyste n " << endl;
                 nn++; }
 n = nn/2;
 h = (b-a)/(double)nn;
 for (int i=1; i <=n-1; i++) s2+=f(a+2*i*h);
 for (int i=0; i <=n-1; i++) s1+=f(a+(2*i+1)*h);
 return h*(f(a)+f(b)+4*s1+2*s2)/3;
}
```

---

Wywołanie funkcji ma postać:

$$\text{simp}(fx, 0, 1, 4)$$

gdzie *fx* jest funkcja podcałkowa, drugi argument jest dolną granicą całkowania (*a*= 0), trzeci argument jest górną granicą całkowania (*b*=1), czwarty argument

jest liczbą przedziałów ( $nn=4$ ).

Program pokazany na listingu 3.6 szacuje naszą całkę metodą Simpsona z opisanymi argumentami.

Listing 3.6. Całkowanie numeryczne, metoda Simpsona, funkcja  $x^3$

---

```
//metoda Simpsona
#include <iostream.h>
#include <conio.h>
double fx(double x)
{return x*x*x;
}
double simp (double (*f)(double), double a, double b, int nn);
int main()
{ cout << "wartosc calki m2 =" << simp(fx,0,1,4)<< endl; //0.25
  getch();
  return 0;
}
double simp (double (*f)(double), double a, double b, int nn)
{int n;
 double s1=0, s2=0, h;
 if (nn%2 != 0) {cout << "nieparzyste n " <<endl;
                 nn++; }
 n = nn/2;
 h = (b-a)/(double)nn;
 for (int i=1; i <=n-1; i++)  s2+=f(a+2*i*h);
 for (int i=0; i <=n-1; i++)  s1+=f(a+(2*i+1)*h);
 return h*(f(a)+f(b)+4*s1+2*s2)/3;
}

```

---

Po uruchomieniu programu mamy wynik:

wartość całki = 0.25

Wiemy, że wartość całki obliczona analitycznie ma taką samą wartość. Metoda Simpsona jest bardzo wydajna i dokładna. Kod funkcji *simp1()* można przedstawić w zwięzłej postaci (listing 3.7).

Listing 3.7. Inna implementacja funkcji simp(), metoda Simpsona

---

```
double simp1 (double (*f)(double), double a, double b, int n)
{ double s=0, h = (b-a)/(double)n;
  for (int i=1; i <=n; i++)

```

```

    { s+=h*(f(a+(i-1)*h)+4*f(a-h/2.0 + i*h) + f(a+i*h))/6.0;
    }
    return s;
}

```

Godna uwagi jest kolejna metoda całkowania numerycznego – metoda Romberga. Jest ona podobna do całkowania metodą trapezów, wykorzystuje ekstrapolację Richardsona. Zainteresowanych czytelników odsyłamy do monografii A.Ralstona (*A First Course In Numerical Analysis*).

Powszechnie uważa się, że metoda nazywana **kwadraturą Gaussa** jest jedną z najmocniejszych metod całkowania numerycznego.

Ta bardzo wydajna metoda całkowania numerycznego, wykorzystuje wielomiany ortogonalne do aproksymacji funkcji. Wybór wielomianów zależy od typu funkcji i granic całkowania. Najczęściej wykorzystywane są wielomiany Legendre’a.

Oszacowanie całki

$$I \approx \int_0^1 x^3 dx$$

przy pomocy kwadratury Gaussa-Legendre’a ma postać:

$$I \approx \frac{b-a}{2} \sum_{k=1}^m w_k f(x_k)$$

Wartości funkcji wyliczane są w punktach:

$$x = \frac{b+a}{2} + \frac{b-a}{2} \xi$$

W powyższych wzorach:

a - jest dolną granicą całkowania, b - jest górną granicą całkowania, m – oznacza liczbę punktów w których funkcja f(x) będzie wyliczana, często m = 4 jest wystarczające, w<sub>k</sub> - są czynnikami wagowymi, ξ<sub>k</sub> - są zerami wielomianu Legendre’a m-tego stopnia:

$$-1 \leq \xi \leq 1$$

Wartości ξ<sub>k</sub> oraz w<sub>k</sub> są stabelaryzowane dla m w zakresie od 2 do 256, do celów praktycznych potrzebujemy tabel z zakresu 2-24.

Dla m = 4, k = 1,2,3,4, potrzebne wartości przedstawia tabela:

- ξ<sub>k</sub> = {-0.8611363116, -0.3399810436, 0.3399810436, 0.8611363116};

- $w_k = \{ 0.3478548451, 0.6521451549, 0.6521451549, 0.3478548451 \}$ ;

Implementacja funkcji do szacowania całek czteropunktową metodą Gaussa-Legendre'a ma postać pokazana na listingu 3.8.

Listing 3.8. Implementacja funkcji gauss\_4(), metoda czteropunktowa

```
double gauss_4 (double (*f)(double), double a, double b)

//a - dolna granica, b - gorna granica calki

{ double const k[4]={-0.8611363116, -0.3399810436, 0.3399810436,
                    0.8611363116};
  double const w[4]={ 0.3478548451, 0.6521451549, 0.6521451549,
                    0.3478548451};
  double xg, s = 0;

  for (int i = 0; i<4; i++)
    { xg=(b+a)/2 + (b-a)*k[i]/2;
      s+=w[i]*f(xg);
    }

  return (b-a)*s/2;
}
```

W celu testowania metody Gaussa oszacujemy całkę

$$I = \int_0^{\pi/2} x^2 \cos x dx$$

przy pomocy kwadratury Gaussa-Legendre'a dla  $m = 4$ .

Prototyp funkcji gauss\_4 wymaga wskaźnika do funkcji:

```
double gauss_4 (double (*f)(double), double a, double b)
```

Postać funkcji podcałkowej:

```
double fx(double x)
{ return x*x*cos(x);
}
```

Wywołanie funkcji Gauss\_4() wymaga jedynie podania granic całkowania:

*gauss\_4(fx,0,pi/2)*

Listing 3.9. Całkowanie numeryczne, czteropunktowa metoda Gaussa

```
//metoda Gaussa, Gauss-Legendre quadrature, m=4
#include <iostream.h>
#include <conio.h>
#include <math.h>
double const pi = 3.1415926535;

double gauss_4 (double (*f)(double), double a, double b);

double fx(double x)
{ return x*x*cos(x);
}

int main()
{
    cout << "wartosc calki = " << gauss_4(fx,0,pi/2)<< endl;
    getch();
    return 0;
}

double gauss_4 (double (*f)(double), double a, double b)
//a - dolna granica, b - gorna granica calki
{ double const k[4]={-0.8611363116, -0.3399810436, 0.3399810436,
                    0.8611363116};
    double const w[4]={ 0.3478548451, 0.6521451549, 0.6521451549,
                       0.3478548451};
    double xg, s = 0;
    for (int i = 0; i<4; i++)
        { xg=(b+a)/2 + (b-a)*k[i]/2;
          s+=w[i]*f(xg);
        }

    return (b-a)*s/2;
}
```

---

Po uruchomieniu programu mamy wynik:

wartość całki = 0.467402

Wartość naszej całki obliczona analitycznie  $I = 0.467401$ . Widzimy doskonałą zgodność oszacowania numerycznego z wartością analityczną.

### 3.3. Rozwiązywanie równań różniczkowych zwyczajnych metodami numerycznymi

Równania różniczkowe odgrywają dużą rolę w opisie zjawisk fizycznych. Niestety w wielu przypadkach rozwiązanie analityczne nie jest znane. Numeryczne metody rozwiązywania równań różniczkowych są stosowane w przypadkach, gdy rozwiązania w postaci analitycznej nie są znane. Ważnym obszarem zastosowania tych metod jest numeryczne modelowanie i symulacja układów i zjawisk dynamicznych. Te techniki są często wykorzystywane w komputerowych symulacjach. W metodach numerycznych zakłada się, że zmienna niezależna występuje w postaci wybranych, dyskretnych wartości. Oczywiście, konsekwencją tego faktu jest otrzymywanie przybliżonych rozwiązań równań różniczkowych.

Równaniem różniczkowym nazywamy równanie funkcyjne, w którym występują pochodne szukanej funkcji  $y = f(x)$ . Stopień największej pochodnej określa rząd równania.

Równanie różniczkowe zwyczajnym rzędu pierwszego ma postać:

$$F(x, y, y') = 0$$

W tym równaniu musi wystąpić pierwsza pochodna  $dy/dx$ , argumenty  $x$  i  $y$  nie muszą jawnie występować. Rozwiązaniem równania różniczkowego rzędu pierwszego (całą równania) nazywamy każdą funkcję postaci:

$$y = \phi(x)$$

która spełnia równanie różniczkowe dla każdej wartości  $x$  z określonego przedziału.

Twierdzenie Cauchy'ego (podstawowe twierdzenie teorii równań różniczkowych):

*Jeśli funkcja  $f(x, y)$  jest ciągła w otoczeniu punktu  $(x_0, y_0)$ , to istnieje przynajmniej jedno rozwiązanie równania*

$$y' = f(x, y)$$

które przy  $x = x_0$  przybiera wartość  $y = y_0$ , a ponadto w pewnym otoczeniu punktu  $x_0$  jest ono oznaczone i ciągłe.

Równanie różniczkowe może być rozwiązane analitycznie. Pokażemy rozwiązanie analityczne prostego równania różniczkowego postaci:

$$\frac{dy}{dx} = 2y$$

Rozwiążemy to równanie metodą rozdzielania zmiennych.

Przy pomocy przekształceń:

$$\frac{1}{2y} dy = dx \quad \Rightarrow \quad \int \frac{1}{2y} dy = \int dx$$

pamiętając, że:

$$\int \frac{1}{x} dx = \ln|c| + C \quad \Rightarrow \quad \log_a x = y, \quad \ln x = \log_e x$$

otrzymujemy rozwiązanie:

$$y = Ce^{2x}$$

gdzie  $C$  jest stałą.

Równanie różniczkowe w praktyce w niewielu prostych przypadkach może być rozwiązane analitycznie. Aby otrzymać rozwiązanie, pozostają nam wyłącznie metody numeryczne. Numeryczne rozwiązywanie równań różniczkowych jest potężną techniką. Niestety nigdy nie można być pewnym wyniku, każde rozwiązanie musi być dokładnie testowane i sprawdzone. Teoria dostarcza nam wiele metod rozwiązywania równań różniczkowych, żadna nie zapewni, że rozwiązanie będzie stabilne. W grafice komputerowej ważna jest także wydajność, konkretnie szybkość otrzymywania rozwiązania.

Równania różniczkowe dzielimy na dwie klasy:

- z warunkami początkowymi (*initial value problems*)
- z warunkami brzegowymi (*boundary value problems*)

W praktyce programowania gier mamy do czynienia z pierwszym typem, tzn. z **równaniami różniczkowymi z warunkami początkowymi**.

Najpierw rozważymy równania różniczkowe z warunkami początkowymi.

W równaniach tego typu, warunki są wyspecyfikowane tylko dla jednej wartości zmiennej niezależnej. Te warunki są nazywane warunkami początkowymi (ang. *initial conditions*) niezależnie od tego czy są wyspecyfikowane dla zmiennej niezależnej równej zero. Typowe równanie tego typu może mieć postać (jest to opis oscylatora harmonicznego):

$$A \frac{d^2y}{dt^2} + B \frac{dy}{dt} + cy = g(t)$$

Z warunkami początkowymi:

$$y(0) = y_0, \quad \frac{dy}{dt}(0) = v_0$$

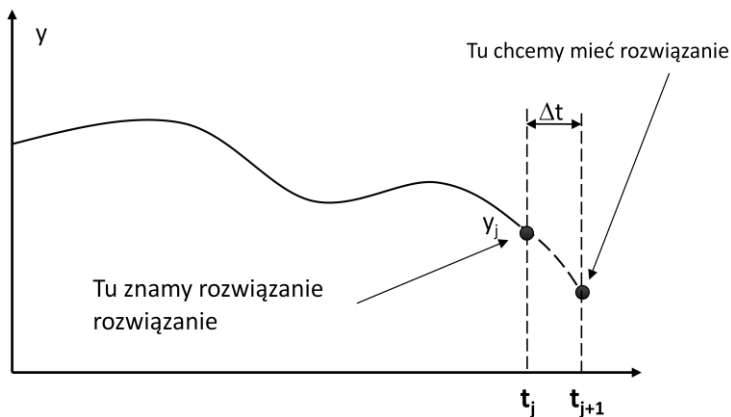
Przy rozwiązywaniu równań różniczkowych:

$$\frac{dy}{dt} = f(y, t)$$

z warunkami początkowymi najczęściej posługujemy się dwoma technikami:

- Metodami jednokrokowymi (ang. *one step type*)
- Metody wielokrokowymi (ang. *multistep type*)

Metody jednokrokowe opierają się na fakcie, że znamy rozwiązanie równania różniczkowego z warunkami początkowymi w przedziale  $0 \leq t \leq t_j$ . W tej metodzie znajdujemy rozwiązanie dla kroku  $t_{j+1} = t_j + \Delta t$  (zobacz rysunek 3.2).



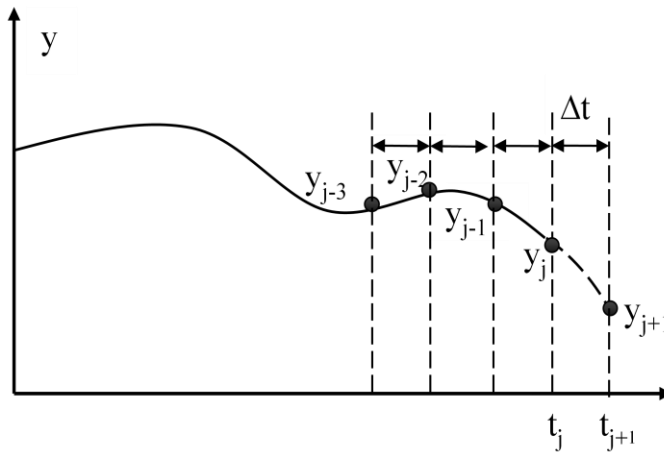
Rys.3.2. Rozwiązania równania różniczkowe typu jednokrokowego. Aby otrzymać rozwiązanie w kroku (j+1) musimy mieć wartości w kroku (j)

Zaletą tej metody jest fakt, że jest ona samo startująca (ang. *self-starting*), ponieważ rozwiązanie jest znajdowane wprost dla wartości od  $t_j$  dla  $t_{j+1}$  bez żądania znajomości wartości dla  $y$  lub  $f(y, t)$  dla  $t < t_j$ . Jeżeli  $t_j = 0$ , i  $y_j = y(0)$ , co mamy wprost z warunków początkowych, nasza metoda może być użyta do znalezienia  $y_{j+1} = y(0 + \Delta t) = y(\Delta t)$  oraz kolejnych wartości dla  $y$ .

W drugiej metodzie (wielokrokowej) musimy mieć informacje dotyczące  $y$  w przedziale  $t < t_j$  (patrz rysunek 3.3).

W rozwiązaniach typu wielokrokowego, rozwiązanie dla  $y_{j+1}$  może wymagać znajomości wartości  $y_j$  oraz wartości  $f(y, t)$  w każdym punkcie  $t_j, t_{j-1}, t_{j-2}$  i  $t_{j-3}$ . Za wyjątkiem najprostszych równań różniczkowych, metoda wielokrokowa nie jest samo startująca i musimy wykorzystać inne techniki aby otrzymać rozwiązania dla pierwszych kilku kroków w procesie rozwiązywania równania (korzystamy także z warunków początkowych).





Rys.3.3. Rozwiązania równania różniczkowe typu jednokrokowego. Aby otrzymać rozwiązanie w kroku  $(j+1)$  musimy mieć wartości w krokach  $(j-3)$ ,  $(j-2)$ ,  $(j-1)$  oraz  $(j)$

Istnieje wiele metod rozwiązywania równań różniczkowych pierwszego rzędu z warunkami początkowymi. W grafice komputerowej najpopularniejsze metody to:

- Metoda Eulera
- Metoda Runge-Kutty
- Metoda Verleta

#### Metoda Eulera.

Rozważmy równanie różniczkowe pierwszego rzędu z warunkami początkowymi:

$$\frac{dy}{dt} = f(y, t), \quad y(0) = y_0$$

Intuicyjnie, proste numeryczne rozwiązanie równania polega na zastąpieniu różniczek różnicami:

$$\frac{y_{j+1} - y_j}{\Delta t} = f(y_j, t_j)$$

Przekształcając to wyrażenie otrzymujemy formułę Eulera (jest tak zwana metoda Eulera):

$$y_{j+1} = y_j + (\Delta t)f(y_j, t_j) \quad (1)$$

Otrzymaliśmy następujący schemat rozwiązywania równania różniczkowego: znając warunki początkowe iterujemy po  $t$ , zaczynając od  $t=0$ , korzystając z równania (1) wyliczamy wartości  $y$  dla każdej nowej wartości  $t$ .

Ponieważ obliczenia możemy wykonać od  $t = 0$ , metoda jest samo-startująca. Zastosujemy schemat Eulera do rozwiązania równania różniczkowego. Rozwiążemy następujące równanie różniczkowe:

$$\frac{dy}{dt} - y^2 = 0 \quad , \quad y(0) = 1$$

Znane nam rozwiązanie analityczne ma następującą postać:

$$y = \frac{1}{1 + t}$$

Przekształcamy równanie wyjściowe do postaci:

$$\frac{dy}{dt} = -y^2, \quad f(y_j, t_j) = -y^2$$

To zadanie może być rozwiązane przy pomocy rekurencyjnej formuły Eulera:

$$y_{j+1} = y_j + \Delta t(-y_j^2) \quad (2)$$

Startując z  $t = 0$  ( $j = 0$ ) oraz wybierając krok czasowy  $\Delta t = 0.1$  możemy wyliczyć wartość  $y$  w punkcie  $t=0.1$ :

$$y_1 = 1 + (0.1)(-1^2) = 0.9$$

Dokładne rozwiązanie w tym punkcie wynosi

$$y(0.1) = \frac{1}{1 + 0.1} = 0.9090909$$

Dla schematu algorytmicznego (2) możemy napisać odpowiedni program komputerowy (rozwiązujący równania różniczkowe metodą Eulera) generujący wartości  $y$  dla ustalonych kroków iteracyjnych.

---

Listing 3.10. Funkcja implementująca metodę Eulera

```
void Euler (double (*f)(double), double dt, double yo, int n)
{ double y, yt;
  y = yo;
  for (int i=1; i <=n; i++)
    { y = y+dt*f(y);   printf("\n t= %4.2f, y= %8.4f",i*dt,y);
    }
}
```

---

Funkcja **Euler()** wymaga: wskaźnika do funkcji  $f()$ , wartości kroku czasowego  $dt$ , wartości początkowej  $y_0$  oraz liczby iteracji  $n$ .

Kompletny program pokazany jest na listingu 3.11.

Listing 3.11. Rozwiązywanie równań różniczkowych, metoda Eulera

```
// Metoda Eulera, rownania rozniczkowe
#include <conio.h>
#include<stdio.h>

void Euler (double (*f)(double), double dt, double yo, int n);
double fx(double y)
{ return -y*y;
}

int main()
{ Euler (fx,0.1,1,10);
  getch();
  return 0;
}

void Euler (double (*f)(double), double dt, double yo, int n)
{ double y, yt;
  y = yo;
  for (int i=1; i <=n; i++)
    { y = y+dt*f(y);   printf("\n t= %4.2f, y= %8.4f",i*dt,y);
    }
}
```

W celu testowania metody Eulera rozwiążemy numerycznie równanie:

$$\frac{dy}{dt} = -y^2, \quad y(0) = 1$$

Prototyp funkcji **Euler()** wymaga wskaźnika do funkcji:

```
void Euler (double (*f)(double), double dt, double yo, int n)
```

Postać funkcji  $f(y,t)$  jest następująca:

```
double fx(double y)
{ return -y*y;
}
```

Wywołanie funkcji **Euler()** wymaga podania nazwy funkcji, kroku iteracji, wartości początkowej oraz liczby iteracji:

*Euler (fx,0.1,1,10);*

Numeryczne rozwiązanie pokazane jest w tabeli 3.1, dla porównania pokazano także wartości dokładne.

Tabela.3.1. Numeryczne rozwiązania równania różniczkowego metodą Eulera oraz rozwiązania analityczne

$i*dt$	$y$ (numeryczne)	$y$ (dokładne)
0.10	0.9000	0.9091
0.20	0.8190	0.8333
0.30	0.7519	0.7692
0.40	0.6954	0.7143
0.50	0.6470	0.6667
0.60	0.6052	0.6250
0.70	0.5685	0.5882
0.80	0.5362	0.5556
0.90	0.5075	0.5263
1.00	0.4817	0.5000

Widzimy, że metoda Eulera technicznie jest bardzo prosta, ale można zauważyć, że nie jest to metoda przesadnie dokładna. Jest rzeczą wiadomą, że metoda Eulera jest praktycznie niedokładna i rzadko bywa używana w praktyce. W grafice komputerowej, gdzie dokładność obliczeń nie jest sprawą kluczową, ze względu na szybkość obliczeń, metoda Eulera czasami jest stosowana.

### Metoda Runge-Kutty.

Wzory typu Runge-Kutty są najczęściej używane do rozwiązywania równań różniczkowych zwyczajnych. Zalety metody są następujące:

- jest łatwa do zakodowania
- jest zazwyczaj stabilne
- krok metody może być zmieniany
- jest samo-startująca

W metodzie zakładamy, że znamy  $y(x_n)$  i chcemy wyznaczyć przybliżenie  $y_{n+1}$  wartości  $y(x_n+h)$ . Metoda polega na obliczeniu wartości  $f(x,y)$  w pewnych szczególnie dobranych punktach, leżących w pobliżu krzywej rozwiązania w przedziale  $(x_n, x_n+h)$  i zastosowaniu odpowiednio dobranego wzoru, zbudowanego z kombinacji tych wartości, tak aby dobrze oszacować przyrost  $y_{n+1} - y_n$ .

Wyróżniamy metody  $n$ -tego rzędu. Rząd określa ile razy obliczamy funkcję  $f(x,y)$ . Najczęściej używana jest formuła Runge-Kutty czwartego rzędu, ale dobre wyniki daje też metoda drugiego rzędu.

Wzory metody Runge-Kutty czwartego rzędu mają postać:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (3)$$

Czynniki  $k$ -te mają następujące znaczenie ( $h$  jest krokiem iteracji)

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \\ k_4 &= hf(x_n + h, y_n + k_3) \end{aligned}$$

Funkcja rozwiązująca równanie różniczkowe metodą Runge-Kutty czwartego rzędu ma postać:

Listing 3.12. Funkcja implementująca metodę Runge-Kutty

---

```
//Metoda Runge-Kutty czwartego rzędu
```

```
void RK_4(double (*f)(double), double dt, double yo, double n)
{ double t,y,k1,k2,k3,k4;
  y=yo;
  for (t=0; t < n; t=t+dt)
  { printf("\n t= %4.2f t y= %8.6f ",t,y);
    k1=f(y)*dt;
    k2=f(y+0.5*k1)*dt;
    k3=f(y+0.5*k2)*dt;
    k4=f(y+k3)*dt;
    y=y+(1/6.0)*(k1+2.0*k2+2.0*k3+k4);
  }
}
```

---

Funkcja **RK\_4()** wymaga: wskaźnika do funkcji  $f()$ , wartości kroku czasowego  $dt$ , wartości początkowej  $y_0$  oraz końcowej wartości  $t$ .

Ponownie rozwiążemy równanie różniczkowe

$$\frac{dy}{dt} = -y^2, \quad y(0) = 1$$

Tym razem zastosujemy metodę Runge-Kutty. Zgodnie ze schematem Runge-Kutty rekurencyjna formuła czwartego rzędu ma postać:

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

Kompletny program rozwiązujący numerycznie równanie różniczkowe metodą Runge-Kutty czwartego rzędu pokazany jest na listingu 3.13.

Listing 3.13. równań różniczkowych, metoda Runge-Kutty

---

```
//Metoda Runge-Kutty 4 rzędu, rownania rozniczkowe
#include<stdio.h>
#include <conio.h>

double fx(double y)
{ return -y*y;
}
void RK_4 (double (*f)(double), double dt, double yo, double n);

int main()
//      dt yo n
{ RK_4 (fx, 1.0, 1.0, 10.0);
  getch();
  return 0;
}
void RK_4 (double (*f)(double), double dt, double yo, double n)
{
  double t,y,k1,k2,k3,k4;
  y =yo;
  for (t=0; t < n; t=t+dt)
  { printf("\n t= %.2f\t y= %8.6f ",t,y);
    k1=f(y)*dt;
    k2=f(y+0.5*k1)*dt;
    k3=f(y+0.5*k2)*dt;
    k4=f(y+k3)*dt;
    y=y+(1/6.0)*(k1+2.0*k2+2.0*k3+k4);
  }
}
```

---

Postać funkcji  $f(y,t)$  jest następująca:

```
double fx(double y)
{ return -y*y;
}
```

Wywołanie funkcji `RK_4()` wymaga podania nazwy funkcji, kroku iteracji, wartości początkowej oraz liczby iteracji:

```
RK_4 (fx,1.0,1.0,10);
```

Numeryczne rozwiązanie pokazane jest w tabeli 3.2, dla porównania pokazano

także wartości dokładne. Zwracamy uwagę, że krok iteracji jest dość duży i wynosi 1.0.

Tabela.3.2. Numeryczne rozwiązania równania różniczkowego metodą Runge-Kutty czwartego rzędu oraz rozwiązania analityczne

$i*dt$	$y$ (numeryczne)	$y$ (dokładne)
0.00	1.000000	1.000000
1.00	0.485636	0.500000
2.00	0.326900	0.333333
3.00	0.246374	0.250000
4.00	0.197676	0.200000
5.00	0.165051	0.166667
6.00	0.141669	0.142857
7.00	0.124089	0.125000
8.00	0.110391	0.111111
9.00	0.099416	0.100000

W porównaniu z metodą Eulera, w pokazanych obliczeniach zwiększono krok iteracji 10 razy, aby pokazać różnice pomiędzy wartością dokładną i przybliżoną rozwiązania równania różniczkowego. Należy pamiętać, że im krótszy krok iteracji, tym większa dokładność rozwiązania.

#### Metoda Runge-Kutty drugiego rzędu.

Jak już wspominaliśmy, istnieje także metoda Runge-Kutty drugiego rzędu (równanie (4)). Metoda Runge-Kutty drugiego rzędu, zwana też metodą Huena, rzadziej nazywana metodą *MidPoint*, jest mniej dokładna niż metoda czwartego rzędu ale jest szybsza.

Należy wspomnieć, że istnieją także inne formuły rozwiązań typu Runge-Kutty takie jak metody szóstego rzędu a nawet ósmego rzędu, ale te techniki są rzadziej używane.

Należy także pamiętać, że zasadniczo (tak samo jak w innych metodach numerycznych) należy odpowiednio dobierać krok iteracji, im mniejszy krok, tym dokładniejsze są rozwiązania. Ustalenie czy wybrano odpowiedni krok polega na uruchomieniu procesu iteracji z dwoma różnymi krokami (najbardziej popularna technika polega na zmniejszeniu dwa razy kroku w drugim przebiegu).

Wzory metody Runge-Kutty drugiego rzędu mają postać:

$$y_{n+1} = y_n + \frac{1}{2}(k_1 + k_2) \quad (4)$$

gdzie współczynniki mają postać:

$$k_1 = hf(x_n, y_n)$$

$$k_2 = hf \left( x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1 \right)$$

Funkcja rozwiązująca równanie różniczkowe metodą Runge-Kutty drugiego rzędu ma postać:

Listing 3.14. Funkcja implementująca metodę Runge-Kutty

```
// Metoda Runge-Kutty drugiego rzędu

void RK_2 (double (*f)(double), double dt, double yo, double n)
{
    double t,y,k1,k2;
    y = yo;
    for (t=0; t<n; t=t+dt)
        { printf("\n t= %4.2f  y= %6.4f ",t,y);
          k1=f(y)*dt;
          k2=f(y+0.5*k1)*dt;
          y=y+k2;
        }
}
```

Funkcja **RK\_2()** wymaga: wskaźnika do funkcji **f()**, wartości kroku czasowego **dt**, wartości początkowej **y0** oraz końcowej wartości **t**.  
Ponownie rozwiążemy nasze różniczkowe

$$\frac{dy}{dt} = -y^2, \quad y(0) = 1$$

Kompletny program rozwiązujący numerycznie równanie różniczkowe metodą Runge-Kutty drugiego rzędu pokazany jest na listingu 3.15.

Postać funkcji **f(y,t)** jest następująca:

```
double fx(double y)
{ return -y*y;
}
```

Wywołanie funkcji **RK\_2()** wymaga podania nazwy funkcji, kroku iteracji, wartości początkowej oraz końcowej wartości przedziału:

```
RK_2 (fx,0.1,1.0,1.0);
```

Numeryczne rozwiązanie pokazane jest w tabeli 3.3 oraz w tabeli 3.4, dla porównania pokazano także wartości dokładne. Zwracamy uwagę, że jeżeli krok iteracji jest dość mały (w naszym przykładzie wynosi on 0.1) to otrzymujemy



dobrą zgodność rozwiązania numerycznego z dokładnym. W przypadku, gdy zwiększymy krok (w pokazanym przykładzie do 10.0) dokładność rozwiązania numerycznego jest znacznie mniejsza.

Listing 3.15. równań różniczkowych, metoda Runge-Kutty

---

```
//Metoda Runge-Kutty 2 rzędu, rownania rozniczkowe
#include<stdio.h>
#include <conio.h>
double fx(double y)
{ return -y*y ;
}
void RK_2 (double (*)(double), double dt, double yo, double n);

int main()
//      dt yo n
{ RK_2 (fx, 0.1, 1.0, 1.0);
  getche();
  return 0;
}
void RK_2 (double (*)(double), double dt, double yo, double n)
{ double t,y,k1,k2;
  y = yo;
  for (t=0; t<n; t=t+dt)
  { printf("\n t= %4.2f y= %6.4f ",t,y);
    k1=f(y)*dt;
    k2=f(y+0.5*k1)*dt;
    y=y+k2;
  }
}
```

---

Analizując dane z tabeli 3.3 widzimy, że metoda Runge-Kutty drugiego rzędu z małym krokiem ( $dt = 0.1$ ) daje dość dokładne rozwiązania w przedziale  $(0.0, 0.90)$ .

Tabela.3.3. Numeryczne rozwiązania równania różniczkowego metodą Runge-Kutty drugiego rzędu oraz rozwiązania analityczne,  $dt = 0.1$ ,  $n = 1.0$

$i*dt$	$y$ (numeryczne)	$y$ (dokładne)
0.00	1.0000	1.0000
0.10	0.9098	0.9091
0.20	0.8343	0.8333
0.30	0.7704	0.7692
0.40	0.7155	0.7143
0.50	0.6679	0.6667
0.60	0.6263	0.6250

0.70	0.5895	0.5882
0.80	0.5567	0.5556
0.90	0.5274	0.5263
1.00	0.5011	0.5000

Analiza tabeli 3.4.(duży krok iteracyjny) skłania nas do wniosku, że metoda Runge-Kutty w przypadku stosowania dużego kroku ( $dt = 1.0$ ) w zakresie (0.0, 9.0) daje wyniki obarczone zbyt dużym błędem.

Tabela.3.4. Numeryczne rozwiązania równania różniczkowego metodą Runge-Kutty drugiego rzędu oraz rozwiązania analityczne,  $dt = 1.0$ ,  $n = 10.0$

$i*dt$	$y$ (numeryczne)	$y$ (dokładne)
0.00	1.0000	1.000000
1.00	0.7500	0.500000
2.00	0.5303	0.333333
3.00	0.3784	0.250000
4.00	0.2843	0.200000
5.00	0.2248	0.166667
6.00	0.1850	0.142857
7.00	0.1568	0.125000
8.00	0.1359	0.111111
9.00	0.1199	0.100000

---

# ROZDZIAŁ 4

## RUCH OBIEKTÓW W POLU GRAWITACYJNYM

---

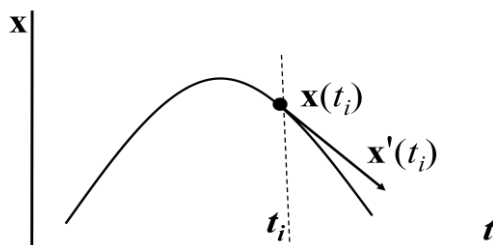
4.1. Równania ruchu .....	76
4.2. Równanie ruchu ze stałym przyspieszeniem.....	78
4.3. Rzut pionowy .....	79
4.4. Rzut poziomy i ukośny.....	83

---

## 4.1. Równania ruchu

Ruch ciał polega na zmianie ich położenia w czasie. Określenie położenia wymaga ustalenia układu współrzędnych. Zjawiska fizyczne modelujemy w przestrzeni trójwymiarowej, co oznacza, że położenie punktu materialnego opisane jest trzema współzrędnymi ( $x, y, z$ ).

Gdy rozpatrujemy wybraną współzrędną na przykład  $x$ , to funkcji położenia  $x(t)$ , opisuje jak  $x$  zmienia się, gdy zmienia się czas  $t$  (rys. 4.1).



Rys.4.1. Zmiana położenie punktu w czasie, funkcja  $x(t)$

W animacji komputerowej istotna jest znajomość prędkości punktu materialnego. Jeżeli dysponujemy funkcją położenia, to także znamy prędkość, ponieważ prędkość jest pochodna położenia po czasie:

$$v(t) = x'(t) = \frac{dx}{dt} = \dot{x}$$

Przypominamy, że symbole  $dx/dt$ ,  $x'$  oraz  $\dot{x}$  oznaczają pochodne. Znając funkcję położenia  $x(t)$  mamy także określone przyspieszenie punktu materialnego  $a$ , jest to pierwsza pochodna prędkości  $v$  po czasie  $t$ , a także druga pochodna położenia po czasie:

$$a(t) = v'(t) = \frac{dv}{dt} = \dot{v}$$

$$a(t) = x''(t) = \frac{d^2x}{dt^2} = \ddot{x}$$

Formalnie ruch ciał w przestrzeni trójwymiarowej opisuje wektor położenia  $\mathbf{r}(t)$ :

$$\mathbf{r}(t) = [x(t), y(t), z(t)]$$

Wektor prędkości ma postać;

$$\mathbf{v}(t) = \dot{\mathbf{x}} = [\dot{x}(t), \dot{y}(t), \dot{z}(t)]$$

a wektor przyspieszenia wyrażamy następująco:

$$\mathbf{a}(t) = \ddot{\mathbf{r}}(t) = [\ddot{x}(t), \ddot{y}(t), \ddot{z}(t)]$$

Na ciała mogą działać siły. Należy pamiętać, że siła nie jest odpowiedzialna za ruch – siła powoduje zmianę ruchu (także zmianę prędkości). Przypominamy, że zgodnie z drugim prawem Newtona:

$$a = \frac{F}{m}$$

wartość wektora przyspieszenia jest proporcjonalna do wartości siły. Znając charakter sił działających na ciało możemy sformułować równanie ruchu w ogólnej postaci:

$$\ddot{\mathbf{r}}(t) = \frac{1}{m} F(\mathbf{r}, \dot{\mathbf{r}}; \boldsymbol{\alpha}; t)$$

gdzie  $\mathbf{m}$  jest masą ciała, a  $\boldsymbol{\alpha}$  oznacza inne istotne parametry ruchu.

Przykładem prostego równania ruchu może być ruch, w którym na ciało o masie  $\mathbf{m}$  działa stała siła:

$$\ddot{\mathbf{r}}(t) = \frac{1}{m} F = a$$

Możemy scałkować to równanie ruchu w granicach,  $(t_0, t)$  a wtedy otrzymamy wzór na prędkość chwilową:

$$v(t) = v(t_0) + a(t - t_0)$$

gdzie  $v(t_0)$  jest prędkością początkową.

W celu otrzymania wzoru na położenie ciała w chwili  $\mathbf{t}$ , ponownie musimy scałkować to równanie po czasie  $\mathbf{t}$  w granicach  $(t_0, t)$ . W wyniku otrzymamy wzór:

$$r(t) = r(t_0) + v(t_0)(t - t_0) + \frac{a}{2}(t - t_0)^2$$

Jest to ważny wzór pozwalający obliczyć położenie obiektu w chwili  $t = t_0 + \Delta t$ , gdy znamy jego położenie w chwili  $t_0$ .

W animacjach komputerowych musimy określać położenie obiektu w określonych chwilach, wyliczamy położenia w kolejnych krokach iteracji:

$$r(t_{n+1}) = r(t_n) + v(t_n)\Delta t + \frac{a}{2}\Delta t^2$$

Tworząc realistyczne gry komputerowe musimy uwzględnić fizykę, zazwyczaj chodzi nam o realistyczny opis ruchu, konkretnie nasze postulaty są następujące:

- chcemy, aby w naszej grafice, obiekty poruszały się w sposób naturalny
- żądamy by ruch był zgodny z naszym codziennym doświadczeniem
- musimy brać pod uwagę uwarunkowania (ograniczenia) sprzętowe
- przetwarzanie nie może być zbyt kosztowne
- musimy wyznaczyć granice dokładności modelowania i symulacji (korzystając z rachunku różniczkowego)

## 4.2. Równanie ruchu ze stałym przyspieszeniem

W przypadku ruchu z przyspieszeniem stałym, mamy do rozwiązania stosunkowo proste równania ruchu. Modelując tego typu ruch dysponujemy zestawem pięciu użytecznych wzorów:

$$v = v_0 + at \quad (1)$$

$$x - x_0 = v_0 t + \frac{1}{2}at^2 \quad (2)$$

$$v^2 = v_0^2 + 2a(x - x_0) \quad (3)$$

$$x - x_0 = \frac{1}{2}(v_0 + v)t \quad (4)$$

$$x - x_0 = vt - \frac{1}{2}at^2 \quad (5)$$

gdzie  $v_0$  – prędkość w chwili  $t = 0$ ,  $v$  – prędkość w chwili późniejszej  $t$ ,  $a$  – stałe przyspieszenie,  $x_0$  – położenie punktu w chwili  $t = 0$ ,  $x$  – położenie punktu w chwili późniejszej  $t$ .

Przyspieszenie obliczamy z wzoru:

$$a = \frac{dv}{dt} = \frac{d}{dt} \left( \frac{dx}{dt} \right) = \frac{d^2x}{dt^2}$$

Podstawowe równania ruchu ze stałym przyspieszeniem można otrzymać całkując równanie definiujące przyspieszenie:

$$\begin{aligned} dv &= a dt \\ \int dv &= \int a dt \end{aligned}$$

Po scałkowaniu otrzymujemy;

$$v = at + C$$

Analogicznie otrzymujemy wzór na drogę:

$$\begin{aligned} dx &= v dt \\ \int dx &= \int v dt \\ \int dx &= \int (v_0 + at) dt \\ \int dx &= v_0 \int dt + a \int t dt \end{aligned}$$

Ostatecznie mamy wzór na położenie:

$$x = v_0 t + \frac{1}{2} at^2 + C'$$

### 4.3. Rzut pionowy

Rozważmy następujący przykład związany z ruchem w polu grawitacyjnym Ziemi. Wyliczmy wybrane wielkości, gdy kamień został rzucony pionowo do góry z prędkością początkową 12 m/s. W rozważaniach pomijamy opór powietrza. Chcemy otrzymać odpowiedź na następujące pytania:

- ile minie sekund do momentu osiągnięcia maksymalnej wysokości?
- ile minie sekund do osiągnięcia wysokości 5 metrów?
- jaka jest maksymalna wysokość rzutu kamienia?

Tego typu proste zadania można rozwiązywać posługując się rozwiązaniami analitycznymi (dokładne rozwiązania).

Rozpatrywany ruch jest ruchem ze stałym przyspieszeniem,  $a = -g$ , gdzie  $g$  jest przyspieszeniem ziemskim,  $g = 9.81 \text{ m/s}^2$ . Taki ruch opisany jest pokazanym uprzednio zestawem pięciu równań.

Rozwiązując zadanie, najpierw wyliczymy ile minie sekund do momentu osiągnięcia maksymalnej wysokości. Na podstawie wzoru nr (1) oraz wiedząc, że  $v = 0$  w najwyższym punkcie obliczamy czas:

$$t = \frac{v - v_0}{a} = \frac{0 - 12}{-9.81} = 1.22$$

Widzimy, że potrzeba 1.22 sekundy, aby osiągnąć maksimum rzutu. Kolejno obliczamy maksymalną wysokość rzutu kamienia.

Na podstawie wzoru (3) oraz faktu, że punkt wyrzucenia kamienia  $x_0 = 0$ , oraz że  $v = 0$  gdy kamień osiągnie maksymalną wysokość, maksymalna wysokość wynosi:

$$x = \frac{v^2 - v_0^2}{2a} = \frac{0 - 12^2}{2(-9.81)} = 7.33$$

Kamień wzniesie się na wysokość 7.33 m.

Na koniec chcemy ustalić ile minie sekund do osiągnięcia wysokości 5 metrów? Korzystamy z równania (2), oraz wiedząc, że  $x_0 = 0$ , oraz  $v = 0$ ,  $x = 2$  m, mamy:

$$x = v_0 t - \frac{1}{2} g t^2$$

Po wstawieniu danych liczbowych:

$$5 = 12t - \frac{1}{2}(9.81)t^2$$

i uporządkowaniu, otrzymujemy równanie kwadratowe:

$$4.9t^2 - 12t + 5 = 0$$

Po rozwiązaniu mamy wartości  $t = 0.52$  s oraz  $t = 1.9$  s.

Nie zawsze można otrzymać rozwiązanie zadania w postaci analitycznej. W takim przypadku musimy wykonać symulacje komputerowe. Aby pokazać metodę, posłużymy się ponownie rzutem pionowym do góry, przyjmujemy, że prędkość początkowa kamienia wyniosła 12 m/s. Będziemy mogli także wykazać poprawność rozwiązania przybliżonego (otrzymanego z symulacji). Kamień rzucony w górę porusza się ze stałym przyspieszeniem,  $\mathbf{a} = -\mathbf{g}$ . W takim ruchu prędkość maleje, w punkcie największego wzniesienia prędkość kamienia  $v = 0$ . Do opisanie ruchu wykorzystujemy zależności:

$$v = v_0 + at$$

$$x - x_0 = \frac{1}{2}(v_0 + v)t$$



Mamy następujące zadanie do rozwiązania: napisać program symulujący ruch kamienia rzuconego pionowo do góry z daną prędkością początkową.

Dla kolejnych wartości upływającego czasu, z chwilą początkową  $t = 0$ , w kolejnych krokach czasowych (z ustalonym przyrostem czasu) obliczamy prędkość obiektu i przebytą drogę. Obliczenia prowadzimy do momentu, gdy prędkość obiektu spadnie do zera. Schemat obliczeń ma następującą postać:

$$t_{i+1} = t_i + dt$$

$$v_{i+1} = v_0 - gt_{i+1}$$

$$s_{i+1} = \frac{1}{2}(v_0 + v)t_{i+1}$$

W kolejnych momentach czasowych obliczamy prędkość i przebytą drogę obiektu, aż do chwili, gdy jego prędkość osiągnie zero. W systemach komputerowych unikamy porównywania liczb zmiennoprzecinkowych z zerem, w zamian wykorzystujemy liczbę dostatecznie małą, oznaczamy ją, jako zmienna *eps*, wyznacza on dokładność obliczeń (obok kroku czasowego *dt*).

W programie kroki iteracyjne wykonuje funkcja o nazwie *simulate()*, której prototyp ma postać:

***double simulate (double (\*f)(double), double v0, double dt, double eps);***

Funkcja *simulate()* otrzymuje wskaźnik do funkcji  $f = gt$ , prędkość początkową, wartość kroku czasowego oraz wartość *eps*.

Listing 4.1. implementacja funkcji *fx()* oraz *simulate()*

---

```
double fx(double t)
{ // v = - gt, m/s, predkosc w ruchu opoznionym
  // g = 9.81 m/s*s, przyspieszenie ziemskie
  return -9.81*t;
}

double simulate (double (*f)(double), double vo, double dt, double eps)
{ double v=vo, s=0, t=0;
  while(v > eps)
  { printf("\n t= %5.2f \t v= %5.2f \t s= %5.2f ", t,v,s);
    t = t + dt;
    v = vo + f(t);
    s = 0.5 * (vo + v) * t;
  }
  return t;
}

```

---

Wykorzystując opracowane funkcje, w wyniku otrzymujemy wartości prędkości obiektu oraz przebytą drogę w funkcji czasu.

Listing 4.2. Modelowanie ruchu obiektu, rzut pionowy

```
// włączyć definicje fx() oraz simulate()
#include <stdio.h>
#include <conio.h>

double fx(double t);

double simulate (double (*f)(double), double vo, double dt, double eps);

int main()
{ //      vo dt eps
  simulate(fx, 12.0, 0.1, 0.01);
  getche();
  return 0;
}
```

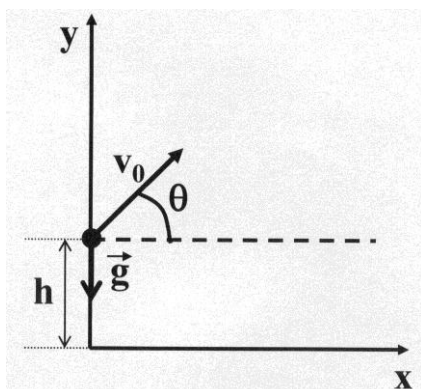
W wyniku działania programu (dla kroku czasowego 0.1 sekundy) otrzymujemy dane pokazane w tabeli 4.1. Maksimum wysokości obiekt osiągnie po 1.2 sekundy i wzniesie się na wysokość 7.34 m. Na dokładność obliczeń wpływa krok czasowy i ustalony błąd (w naszym przypadku zakładamy błąd na poziomie 1 procenta ( $\text{eps} = 0.01$ )). Gdy krok czasowy zmniejszymy do 0.01 sekundy otrzymamy następujący wynik:  $t = 1.22$ ,  $v = 0.03$ ,  $s = 7.34$ .

Tabela 4.1. Droga i czas w rzucie pionowym, program 4.2

t(czas)	v(prędkość)	s(droga)
0.00	12.00	0.00
0.10	11.02	1.15
0.20	10.04	2.20
0.30	9.06	3.16
0.40	8.08	4.02
0.50	7.09	4.77
0.60	6.11	5.43
0.70	5.13	6.00
0.80	4.15	6.46
0.90	3.17	6.83
1.00	2.19	7.09
1.10	1.21	7.26
1.20	0.23	7.34

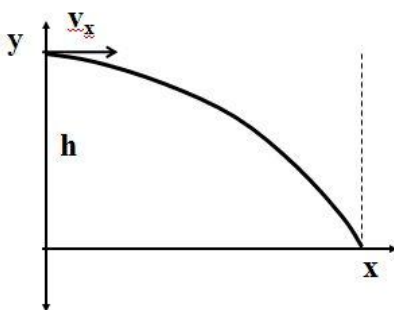
#### 4.4. Rzut poziomy i ukośny

W polu grawitacyjnym Ziemi ruch obiektów jest ruchem krzywoliniowym. Pod wpływem siły ciężkości ciało będzie spadało (rys.4.1 oraz 4.2). Wyróżniamy ruch poziomy i ruch ukośny. Rzut poziomy jest to ruch, w którym ciało nadajemy prędkość skierowaną równoległe do poziomu. Przykładem takiego ruchu może być ruch pocisku wystrzelonego z poziomo ustawionej strzelby. W ogólnym przypadku ruch taki odbywa się w przestrzeni trójwymiarowej, często jednak dla potrzeb grafiki komputerowej można rozważać kinetykę punktu materialnego w przestrzeni dwuwymiarowej.



Rys.4.2. Ruch obiektu w polu grawitacyjnym,  $v_0$  jest to prędkość początkowa, wektor  $v_0$  tworzy kąt  $\theta$  z dodatnim kierunkiem osi  $x$ .

W **rzucie poziomym** (rys. 4.3) kąt  $\theta$  ma wartość zero, zwrot wektora prędkości  $v_0$  w chwili początkowej jest skierowany wzdłuż osi  $x$ , ciało znajduje się na wysokości  $y = h$ .



Rys. 4.3. Rzut poziomy, prędkość początkowa wynosi  $v_x$

W celu wyznaczenia kształtu toru obiektu w rzucie poziomym musimy rozwiązać równania ruchu. Wiemy, że ruch w poziomie jest niezależny od ruchu w pionie.

Wiemy także, że w rzucie ukośnym parametryczne równania ruchu są zapisane

następująco:

$$x(t) = v_x t$$

$$y(t) = h - \frac{1}{2} g t^2$$

Naszym celem jest wyznaczenie toru obiektu, któremu nadano prędkość początkową  $v_x$  i skierowana pod kątem  $0^0$  do poziomu. Obliczenia możemy prowadzić w przedziale czasu  $(t_0, t)$  otrzymując wartości położenia, współrzędne  $x$  i  $y$ . Zasadnicze obliczenia przeprowadza funkcja *simulate()*:

***simulate(fx, v<sub>x</sub>, h, dt, eps);***

gdzie  $v_x$  jest prędkością w kierunku poziomym,  $h$  jest wysokością na poziomie,  $dt$  jest krokiem iteracji,  $eps$  określa dokładność obliczeń. Cały program przedstawiony jest na listingu 4.4

Listing 4.3. Modelowanie ruchu obiektu, rzut poziomy

---

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
double fx(double t);
double simulate(double (*f)(double), double vo, double h, double dt, double eps);
int main()
{
    //      vo  h  dt  eps
    simulate(fx, 55.0, 500.0, 1.0, 0.1);
    getch();
    return 0;
}
double fx(double t)
{
    return 9.81*t;
}
double simulate(double (*f)(double), double vo, double h, double dt, double eps)
{
    double v, vy = 0.0;
    double x=0.0, y = h;
    double t = 0.0;
    printf("\n czas |t| x |t| y |t| vy |t| v"|, t,x,y,vy);
    while (y > -eps)
    {
        printf("\n %5.2ft |%5.2ft |%5.2ft |%6.2ft |%6.2ft"|, t,x,y,vy,v);
        t = t + dt;
        x = vo*t;
        y = h - 0.5*t*t*f(t);
        vy = - f(t);
        v=sqrt(vo*vo+vy*vy);
    }
    return t;
}

```

---

Wyniki symulacji rzutu poziomego z parametrami:

```
//      vo  h  dt  eps
simulate(fx, 55.0, 500.0, 1.0, 0.1);
```

pokazane są w tabeli 4.2.

Tabela 4.2. Tor obiektu w rzucie poziomym

czas	x	y	vy	v
0.00	0.00	500.00	0.00	0.00
1.00	55.00	495.10	-9.81	55.87
2.00	110.00	480.38	-19.62	58.39
3.00	165.00	455.86	-29.43	62.38
4.00	220.00	421.52	-39.24	67.56
5.00	275.00	377.38	-49.05	73.69
6.00	330.00	323.42	-58.86	80.56
7.00	385.00	259.65	-68.67	87.98
8.00	440.00	186.08	-78.48	95.83
9.00	495.00	102.69	-88.29	104.02
10.00	550.00	9.50	-98.10	112.47

Symulacja toru obiektu może być zastosowana w celu modelowania zrzutu kapsuły ratunkowej z samolotu. Przyjmijmy, że samolot leci na wysokości 500 metrów z prędkością 55 m/s. Krok symulacji wynosi 1 sekundę. Z tabeli 4.2 wynika, że lot kapsuły od momentu wyrzucenia jej z samolotu do momentu uderzenia w powierzchnię morza wynosi około 10 sekund, kapsuła przeleci odległość 550 metrów. Prędkość kapsuły w momencie uderzenia w wodę wynosi 112 m/s. Jeżeli zmniejszymy krok iteracji  $dt$  do wartości 0.01 sekundy to otrzymamy znacznie dokładniejszy wynik: lot kapsuły będzie trwał 10.09 sekundy, kapsuła przeleci drogę  $x = 554.95$  metrów a prędkość końcowa wyniesie 113.24 m/s.

Rozważając **rzut ukośny** wiemy, że ruch w kierunku poziomym odbywa się bez przyspieszenia, dlatego też składowa pozioma prędkości  $v_x$  obiektu nie ulega zmianie w czasie ruchu i jest równa prędkości początkowej  $v_0$ . Wiemy, że w rzucie ukośnym parametryczne równania ruchu są zapisane następująco:

$$x(t) = v_x t = (v_0 \cos \theta) t$$

$$y(t) = v_y t - \frac{1}{2} g t^2 = (v_0 \sin \theta) t - \frac{1}{2} g t^2$$

Naszym celem jest wyznaczenie toru obiektu, któremu nadano prędkość początkową  $v_0$  i skierowana pod kątem  $\theta$  do poziomu. Obliczenia możemy prowadzić w przedziale czasu  $(t_0, t)$  otrzymując wartości położenia, współrzędne  $x$  i  $y$ .

Zasadnicze obliczenia przeprowadza funkcja `simulate()`:

```
simulate(fx, 20.0, 30.0, 0.1, 1.0);
```

Cały program przedstawiony jest na listingu 4.4

Listing 4.4. Modelowanie ruchu obiektu, rzut ukośny

---

```
#include <stdio.h>
#include<conio.h>
#include <math.h>
#define R 3.1415926/180.0
//dodac funkcje simulate()
double fx(double t);
double simulate(double (*f)(double),double vo,double alfa,double dt,double eps);
int main()
{
    //      vo alfa dt eps
    simulate(fx, 20.0, 30.0, 0.1, 1.0);
    getch();
    return 0;
}
double fx(double t)
{ return 9.81*t;
}
}
```

---

Funkcja `simulate()` otrzymuje następujące parametry: prędkość początkową  $v_0$ , kąt  $\theta$ , krok czasowy  $dt$  oraz warunek stopu  $eps$ . Symulacje możemy prowadzić w dowolnym przedziale czasowym, w naszym przypadku iteracja jest zatrzymywana, gdy wartość współrzędnej  $y$  staje się ujemna (obiekt przebił poziom). Dokładność obliczeń zależy od kroku czasowego  $dt$  i wartości  $eps$ .

Listing 4.4. funkcja `simulate()`, rzut ukośny

---

```
double simulate(double (*f)(double),double vo,double alfa,double dt,double eps)
{ double v=0.0, vx, vx2, vy;
  double x,y;
  double t = 0.0;
  double co = cos(R*alfa);
  double si = sin(R*alfa);
  vx = vo*co;
  vy = vo*si;
  vx2 = vx*vx;
  v=sqrt(vx2+vy*vy);
  printf("\n czas \t x \t y \t vy \t v", t,x,y,vy);
  while (y > -eps)
```

```

{ printf("\n %5.2f\t %5.2f\t %5.2f\t %6.2f\t %6.2f",t,x,y,vy,v);
  t = t + dt;
  x = vo*co*t;
  y = vo*si*t - 0.5*t*f(t);
  vy = vo*si - f(t);
  v=sqrt(vx2+vy*vy);
}
printf("\n vx = %6.2f",vx);
return t;
}

```

W tabeli 4.3 przedstawiono wyniki symulacji dla kroku czasowego  $dt = 0.1$  s, prędkości początkowej 20 m/s, kąta  $\theta = 30^\circ$  oraz  $eps = 0.1$ .

Tabela. 4.3. Tor obiektu w rzucie ukośnym

czas	x	y	vy	v
0.00	0.00	0.00	10.00	20.00
0.10	1.73	0.95	9.02	19.53
0.20	3.46	1.80	8.04	19.09
0.30	5.20	2.56	7.06	18.70
0.40	6.93	3.22	6.08	18.36
0.50	8.66	3.77	5.09	18.05
0.60	10.39	4.23	4.11	17.80
0.70	12.12	4.60	3.13	17.60
0.80	13.86	4.86	2.15	17.45
0.90	15.59	5.03	1.17	17.36
1.00	17.32	5.09	0.19	17.32
1.10	19.05	5.06	-0.79	17.34
1.20	20.78	4.94	-1.77	17.41
1.30	22.52	4.71	-2.75	17.54
1.40	24.25	4.39	-3.73	17.72
1.50	25.98	3.96	-4.72	17.95
1.60	27.71	3.44	-5.70	18.23
1.70	29.44	2.82	-6.68	18.56
1.80	31.18	2.11	-7.66	18.94
1.90	32.91	1.29	-8.64	19.36
2.00	34.64	0.38	-9.62	19.81
2.10	36.37	-0.63	-10.60	20.31
vx =	17.32			

Z pokazanych symulacji wynika, że z naszymi danymi:

```

//      vo alfa dt eps
simulate(fx, 20.0, 30.0, 0.1, 1.0);

```

czas trwania rzutu wynosi około 2 sekund, zasięg rzutu wynosi około 35 metrów, maksymalna wysokość równą 5 metrów obiekt osiąga po jednej sekundzie, a prędkość całkowita w tym momencie jest najmniejsza i wynosi około 17.3 m/s.

W programie obliczamy także składową prędkości  $v_y$  oraz prędkość całkowitą  $v$ . Należy mieć na uwadze, że w tych symulacjach nie uwzględniono oporu powietrza. Opór powietrza zdecydowanie potrafi zmienić tor rzuconego obiektu. Jak podają D.Halliday i inni w podręczniku „Podstawy fizyki”, piłka baseballowa poruszająca się z prędkością początkową 44.7 m/s i wybita pod

kątem  $60^0$  ma zasięg w powietrzu równy 98.5 metra, gdyby ruch odbywał się w próżni zasięg wynosiłby 177 metrów! Jak można się przekonać korzystając z naszego programu wyliczony zasięg wynosi dla tego przypadku 176 metrów.

Ze wzorów parametrycznych możemy wyeliminować czas, otrzymamy równanie toru ciała:

$$y = (tg\theta)x - \frac{gx^2}{2(v_0 \cos\theta)^2}$$

Jak widać jest to równanie typu:

$$y = ax + bx^2$$

Otrzymaliśmy równanie paraboli, tor ciała w rzucie ukośnym jest paraboliczny.



---

# ROZDZIAŁ 5

## RUCH OBIEKTÓW Z UWZGLĘDNIENIEM OPORÓW POWIETRZA

---

5.1. Siła oporu .....	90
5.2. Spadek pionowy z uwzględnieniem oporu powietrza .....	94
5.3. Rzut poziomy z uwzględnieniem oporu powietrza .....	98
5.4. Rzut ukośny z uwzględnieniem oporu powietrza.....	101
5.5. Efekt Magnusa i ruch obiektów .....	109

---

## 5.1. Siła oporu

W warunkach naturalnych najczęściej nie mamy do czynienia z ruchem w próżni. Oczywiście obiekty takie jak statki kosmiczne praktycznie poruszają się w próżni, gdy znajdują się na swojej orbicie stacjonarnej, ale jest to wyjątek od reguły. Praktycznie obiekty na Ziemi poruszają się w powietrzu (gaz) lub w cieczy (najczęściej jest to woda). Gdy zachodzi ruch w tych mediach to mamy do czynienia z działaniem siły oporu  $D$ . Opór w gazie nazywany jest **oporem aerodynamicznym**, opór w cieczy nazywany jest **oporem hydrodynamicznym**. Występujące w tych warunkach siły utrudniają ruch (powoduje to spowolnienie ruchu) i są skierowane przeciwnie do kierunku ruchu obiektu. Zasadniczo jest bardzo trudno podać dokładne formuły pozwalające na wyliczenie sił oporu. Jesteśmy zdani na wzory przybliżone lub dane doświadczalne.

W powietrzu, gdy ruch ciała jest dostatecznie szybki, tak że mamy do czynienia z przepływem turbulentnym wartość siły oporu  $D$  może być dana wzorem:

$$D = \frac{1}{2} C \rho S v^2$$

przy czym  $\mathbf{v}$  jest prędkością względną,  $C$  jest doświadczalnie wyznaczanym współczynnikiem oporu aerodynamicznego,  $\rho$  jest gęstością powietrza,  $S$  jest polem przekroju poprzecznego. Współczynnik  $C$  przybiera wartości w zakresie od 0.4 do 1.0, jest w przybliżeniu stały (ale w zasadzie zależy od  $\mathbf{v}$ ). Wykładnik potęgi stojący przy prędkości może przybierać wartości z zakresu od 1 do 5. Możemy znacznie uprościć zagadnienie przyjmując, że opór powietrza dany jest wzorem:

$$D = -kv$$

gdzie  $k$  jest współczynnikiem opisującym opory powietrza (zależy od środowiska i kształtu ciała), a  $\mathbf{v}$  jest prędkością obiektu. Siła oporu zawsze jest skierowana stycznie do toru obiektu. Gdy mamy do czynienia ze spadkiem swobodnym w powietrzu, to siła oporu  $D$  skierowana jest do góry, jej wartość wzrasta w czasie spadku. Siła  $D$  jest przeciwna do skierowanej w dół siły ciężkości  $F_g$ . Druga zasada dynamiki Newtona ma postać:

$$D - F_g = ma$$

gdzie  $m$  jest masą obiektu.

Po ustalonym czasie siły  $D$  i  $F_g$  mogą się zrównoważyć, przyspieszenie będzie równe zeru i prędkość obiektu w takim przypadku nie będzie wzrastała. Obiekt będzie się poruszał ze stałą prędkością – mówimy, że obiekt osiągnął prędkość graniczną.

Możemy wyznaczyć tą prędkość:

$$\frac{1}{2} C \rho S v_t^2 - F_g = 0$$

$$v_t = \sqrt{\frac{2F_g}{C\rho S}}$$

W podręczniku Hallidaya i innych podano wartości prędkości granicznych dla wybranych obiektów. W tabeli 5.1. pokazano te wartości.

Tabela 5.1. Prędkości graniczne (Halliday i inni).

Obiekt	$v_t$ (m/s)
Skoczek bez spadochronu	60
Skoczek ze spadochronem	5
Piłka tenisowa	31
Piłka do koszykówki	20
Piłka pingpongowa	9

Wpływ oporu powietrza na ruch obiektów może być znaczny. Obliczymy prędkość kropli deszczu na poziomie morza, gdy uwzględnimy opór. Niech chmura deszczowa znajduje się na wysokości 1000 metrów. Kropla ma średnicę około 1.5 mm, przy tym rozmiarze jej masa wynosi około 14.17 mg, przyjmijmy za Hallidayem, że współczynnik oporu  $C = 0.6$ . Biorąc pod uwagę, że gęstość wody to  $1000 \text{ kg/m}^3$ , a gęstość powietrza to  $1.2 \text{ kg/m}^3$  możemy obliczyć prędkość graniczną kropli deszczu.

Pamiętając wzór na siłę ciężkości:

$$F_g = m \cdot g$$

oraz, że wzór na prędkość  $v_t$  ma postać:

$$v_t = \sqrt{\frac{2F_g}{C\rho_p S}}$$

wyliczamy prędkość (pole przekroju poprzecznego kropli  $S = \pi R^2$ )

$$\sqrt{\frac{2F_g}{C\rho_p S}} = \sqrt{\frac{2 \cdot 14.17 \cdot 10^{-6} \cdot 9.81}{0.6 \cdot 1.2 \cdot 3.1415 \cdot (1.5 \cdot 10^{-3})^2}} = \sqrt{\frac{278.01}{5.09}} = 7.39$$

Prędkość graniczna kropli deszczu przy powierzchni Ziemi wynosi około 7.4 m/s (27 km/h).

Korzystając ze wzoru na prędkość w spadku swobodnym:

$$v = \sqrt{2hg}$$

oraz wiedząc, że chmura jest na wysokości 1000 metrów, końcowa prędkość kropli, gdyby nie było oporu powietrza, byłaby równa 140 m/s (504 km/h)! Widzimy wyraźnie jak duży wpływ na ruch obiektów ma opór powietrza.

Możemy znacznie uprościć modelowanie ruchu, jeżeli przyjmiemy założenie, że siła oporu powietrza  $D$  jest proporcjonalna do prędkości:

$$D = -kv$$

Znak minus oznacza, że siła oporu jest skierowana przeciwnie do kierunku ruchu. Zwracamy uwagę, że współczynnik proporcjonalności  $k$  ma inną wartość niż współczynnik proporcjonalności  $C$  omawiany poprzednio!

Gdy ciało spada z określonej wysokości w powietrzu to na ciało działają dwie siły – siła grawitacji i siła oporu. Gdy prędkość wzrasta, wzrasta także siła oporu, aż do momentu, gdy te dwie siły się równoważą. Siła wypadkowa staje się równa zero, przyspieszenie ma wartość zerową, prędkość jest stała. Tą prędkość nazywamy prędkością graniczną. Zgodnie z drugą zasadą Newtona równanie ruchu ma postać:

$$mg = kv_{gr}$$

$$v_{gr} = \frac{mg}{k}$$

Możemy rozwiązując równania ruchu wyznaczyć prędkość ciała podczas ruchu pionowego z uwzględnieniem oporu powietrza. Równanie ruchu ma postać:

$$ma = mg - kv$$

$$m \frac{dv}{dt} = mg - kv$$

To równanie różniczkowe ma rozwiązanie analityczne w postaci:

$$v(t) = \frac{mg}{k} \left(1 - e^{-\frac{k}{m}t}\right) = v_{gr} \left(1 - e^{-\frac{k}{m}t}\right) \quad (1)$$

Mamy też proste wyrażenie na przyspieszenie w omawianym ruchu:

$$a(t) = g - \frac{k}{m}v(t)$$

Korzystając z założenia, że siła oporu powietrza jest wprost proporcjonalna do prędkości oraz mając dokładne rozwiązania równania różniczkowego (1), możemy modelować prędkość i przyspieszenie ciała w spadku pionowym z uwzględnieniem oporu powietrza.

#### Listing5.1. Prędkość w spadku pionowym z oporem powietrza

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
double simulate(double m, double c, double dt);
int main()
{
    //      m      c      dt
    simulate( 75.0, 122.0, 0.01);
    getch();
    return 0;
}
double simulate(double m, double c, double dt)
{
    double vy=0.0;    //predkosc
    double ay;        //przyspieszenie
    double t = 0.0;
    double r,r1;
    int n=0;
    double g = 9.81;
    r = c/m;
    r1 = g/r;
    printf("\n czas (s)  vy (m/s)  ay (m/s*s)");
    do
    {
        vy = r1*(1-exp(-(r*t)));
        ay = g - r*vy;
        if (!(n%20))
            printf("\n %8.3f %8.2f %8.2f",t,vy,ay);
        t = t + dt;
        n++;
    } while (t <3.0);
    return t;
}
```

Dla przyjętych parametrów funkcji  $\text{simulate}(75.0, 122.0, 0.01)$ , tzn. masa obiektu wynosi 75 kg, współczynnik oporu jest równy 122 oraz krok czasowy jest równy 0.01 s w tabeli 5.1 pokazano wyniki modelowania. Należy pamiętać, że współczynniki oporu są trudne do wyliczenia, posługujemy się najczęściej parametrami wyznaczonymi doświadczalnie. W literaturze można spotkać, że np. prędkość graniczna skoczka spadochronowego (zależna od masy i współczynnika oporu) może być równa 5 m/s (podręcznik Hallidaya) lub równa około 6 m/s (podręcznik M. Zawackiego, „Fizyka, ćwiczenia praktyczne”). Oznacza to, że współczynnik oporu  $k$  jest w przedziale (120, 150).

Tabela 5.1. Zależność prędkości i przyspieszenia w spadku pionowym z oporem powietrza, rozwiązanie analityczne

czas (s)	$v_y$ (m/s)	$a_y$ (m/s*s)
0.000	0.00	9.81
0.200	1.67	7.09
0.400	2.88	5.12
0.600	3.76	3.70
0.800	4.39	2.67
1.000	4.85	1.93
1.200	5.17	1.39
1.400	5.41	1.01
1.600	5.58	0.73
1.800	5.71	0.52
2.000	5.80	0.38
2.200	5.86	0.27
2.400	5.91	0.20
2.600	5.94	0.14
2.800	5.97	0.10
3.000	5.98	0.07

Z pokazanej tabeli wynika, że spadający obiekt w początkowej fazie porusza się ruchem zmiennym (wzrasta prędkość, przyspieszenie ma wartość niezerową). Po około 2.5 - 3.0 sekundach przyspieszenie spada do zera, ciało osiąga wartość graniczną (około 6 m/s), dalszy spadek jest ruchem jednostajnym.

## 5.2. Spadek pionowy z uwzględnieniem oporu powietrza

Obliczymy tor ruchu oraz inne parametry w przypadku spadku obiektu w polu grawitacyjnym Ziemi z uwzględnieniem oporu powietrza. Tego typu ruch dobrze opisuje ruch skoczka spadochronowego. Ruch skoczka modelowany będzie przy założeniu, że spadochroniarz otworzył spadochron natychmiast (jego prędkość początkowa jest równa zero, otwarcie spadochronu nastąpiło na wysokości  $h$  metrów). Znamy masę spadochroniarza oraz współczynnik oporu powietrza  $c$  (przyjmujemy, że  $c = 20$ ). Zakładamy, że siła oporu  $D$  jest proporcjonalna do kwadratu prędkości  $v$

$$D = cv^2$$

Równanie ruchu ma postać

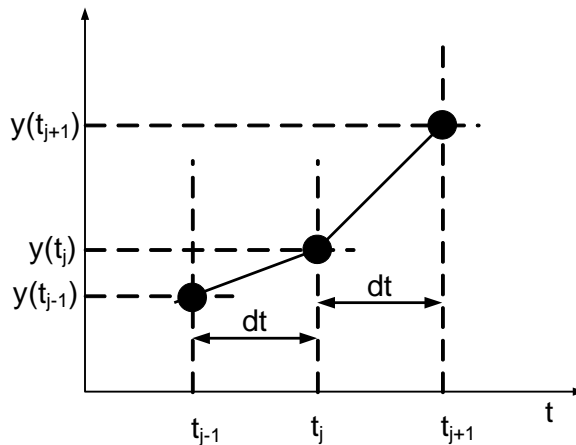
$$ma = -mg + D$$

gdzie  $m$  jest masą,  $a$  - przyspieszeniem ciała,  $g$  - przyspieszeniem ziemskim. W opisie ruchu opadającego obiektu należy wyznaczyć:

- $y(t)$  – zależność przebytej drogi od czasu
- $v(t)$  – zależność prędkości od czasu
- $a(t)$  – zależność przyspieszenia od czasu

Można pokazać (podręcznik M. Zawackiego), że do iteracyjnego wyliczenia położenia  $y(t_{j+1})$  wystarczy znać dwa poprzednie położenia:  $y(t_{j-1})$  oraz  $y(t_j)$ . Schemat obliczeń pokazany jest na rys. 5.1. Wzór iteracyjny do wyznaczenia kolejnych położenia ciała w spadku pionowym z uwzględnieniem oporu powietrza ma postać ( $t$  oznacza czas,  $dt$  oznacza przyrost czasu) :

$$y(t_{j+1}) = 2y(t_j) - y(t_{j-1}) - g(dt)^2 + \frac{c}{m} (y(t_j) - y(t_{j-1}))^2$$



Rys. 5.1. Schemat obliczeń kolejnych położenia obiektu.

Na podstawie wyliczonych w kolejnych krokach położenia możemy otrzymać wzory na obliczenie prędkości:

$$v(t_j) = \frac{y(t_j) - y(t_{j-1})}{dt}$$

oraz przyspieszenia:

$$a(t_j) = -g + \frac{c}{m} v^2(t_j)$$

Na listingu 5.2. pokazany jest program do obliczania numerycznego położenia obiektu, prędkości i przyspieszenia w funkcji czasu w spadku pionowym obiektu z uwzględnieniem oporu powietrza.

Listing5.2. Modelowanie spadku pionowego z oporem powietrza

---

```
#include <stdio.h>
#include<conio.h>
#include <math.h>

double simulate(double h, double m, double c, double dt, double eps);
int main()
{
    //      h      m      c      dt      eps
    simulate( 500.0, 75.0, 20.0, 0.005, 20.0);
    getche();
    return 0;
}

double simulate(double h, double m, double c, double dt, double eps)
{
    double y[3];      //przebyta droga
    double vy, ay;    //predkosc i przyspieszenie
    double t = 0.0, t1;
    double r=c/m;
    int n=0;
    double g = 9.81;
    printf("\n  v graniczna = %8.4f m/s",sqrt(m*g/c));
    y[0] = h;
    y[1] = y[0] - g*dt*dt;
    printf("\n czas (s)  y (m)  v (m/s)  a (m/s*s) ");
    do
    {
        y[2] = 2.0*y[1]- y[0] - g*dt*dt + r*(y[1]-y[0])*(y[1]- y[0]);
        vy = (y[1] - y[0])/dt ;
        ay = (y[2] + y[0] - 2.0*y[1])/(dt*dt);
        if ( !(n%1000) )
            printf("\n %6.3ft %8.2f %8.2f %8.2f",t,y[2],vy, ay);
        y[0]= y[1];
        y[1]= y[2];
        t = t + dt ;
        n++;
    } while (y[2] > -eps); //(t < 2.5);
    return t;
}
```

---



Tabela 5.2. pokazuje wyniki obliczeń numerycznych. Modelowanie ruchu wykonane jest z bardzo małym krokiem czasowym ( $\Delta t = 0.005$  s), w tej sytuacji w tabeli pokazano tylko wybrane obliczenia, dzięki instrukcji:

```
if ( !(n%1000) )
    printf("\n %6.3f\t %8.2f %8.2f %8.2f",t,y[2],vy, ay);
```

gdzie  $n$  jest numerem iteracji. Iteracja przerwana jest w momencie otrzymania wartości drogi mniejszej od  $\epsilon$  (na tym listingu  $\epsilon = 20$  m). Widzimy, że spadek z wysokości 500 metrów trwał około 82 sekund.

Tabela 5.2. Zależność drogi od czasu w spadku pionowym z oporem

v graniczna = 6.0653 m/s			
czas (s)	y (m)	v (m/s)	a (m/s*s)
0.000	500.00	-0.05	-9.81
5.000	472.19	-6.07	-0.00
10.000	441.86	-6.07	0.00
15.000	411.54	-6.07	0.00
20.000	381.21	-6.07	0.00
25.000	350.88	-6.07	0.00
30.000	320.56	-6.07	0.00
35.000	290.23	-6.07	0.00
40.000	259.90	-6.07	0.00
45.000	229.58	-6.07	0.00
50.000	199.25	-6.07	0.00
55.000	168.92	-6.07	0.00
60.000	138.60	-6.07	0.00
65.000	108.27	-6.07	0.00
70.000	77.95	-6.07	0.00
75.000	47.62	-6.07	0.00
80.000	17.29	-6.07	0.00
85.000	-13.03	-6.07	0.00

Z tabeli 5.2 nie można otrzymać zbyt wiele informacji o zależności prędkości i przyspieszenia. Spowodowane jest to drukowaniem wyników z krokiem czasowym 5 sekund. Aby otrzymać bardziej interesujące wyniki należy wykonać obliczenia z wypisywaniem danych z krótszym czasem. W tabeli 5.3 pokazano wyniki obliczeń, gdy iteracje zatrzymano po 2.5 sekundach, wyniki są produkowane co 0.125 sekund. W programie zmieniono warunek stopu:

```
while (y[2] > -eps); //(t < 2.5);
```

Z tak szczegółowych danych wynika, że prędkość ciała wzrastała bardzo szybko a prędkość graniczna została osiągnięta po 2.25 sekundach, po przebyciu około 12 metrów. Podobnie zanalizować możemy przebieg zależności przyspieszenia od czasu – przyspieszenie od początkowej wartości równej  $9.81 \text{ m/s}^2$  zaczyna maleć, aby po około 2.5 sekundy osiągnąć wartość zero. Od tej chwili ruch jest ruchem jednostajnym.

Zauważalne różnice w obu czasach spowodowane są niedokładnością obliczeń.

Tabel 5.3 prędkość i przyspieszeni w spadku pionowym z oporem

v graniczna =		6.0653 m/s	
czas (s)	y (m)	v (m/s)	a (m/s*s)
0.000	500.00	-0.05	-9.81
0.125	499.91	-1.26	-9.39
0.250	499.67	-2.37	-8.31
0.375	499.31	-3.33	-6.86
0.500	498.83	-4.09	-5.34
0.625	498.28	-4.68	-3.98
0.750	497.66	-5.10	-2.86
0.875	497.00	-5.41	-2.01
1.000	496.31	-5.62	-1.39
1.125	495.60	-5.77	-0.95
1.250	494.87	-5.86	-0.64
1.375	494.13	-5.93	-0.43
1.500	493.39	-5.98	-0.29
1.625	492.64	-6.01	-0.19
1.750	491.89	-6.03	-0.13
1.875	491.13	-6.04	-0.09
2.000	490.38	-6.05	-0.06
2.125	489.62	-6.05	-0.04
2.250	488.86	-6.06	-0.03
2.375	488.11	-6.06	-0.02
2.500	487.35	-6.06	-0.01

### 5.3. Rzut poziomy z uwzględnieniem oporu powietrza

W podręcznikach fizyki kinematyka i dynamika obiektów rozpatrywana jest w warunkach, gdy ruch nie jest zakłócany (np. ruch odbywa się w próżni). Dzięki takim założeniom, otrzymujemy przejrzyste formuły opisujące ruch. Niestety, gdy chcemy te wzory zastosować w praktyce, wyniki obliczeń znacznie odbiegają od danych realnych. Poprzednio rozpatrywaliśmy rzut poziomy przy założeniu, że nie istnieje opór powietrza. W rzeczywistości opisując rzut poziomy musimy uwzględnić opór powietrza, gdyż wpływa on znacznie na tor poruszającego się obiektu. Jeżeli chcemy uwzględnić opór powietrza, jak zwykle mamy wybór – możemy założyć, że siła oporu jest albo proporcjonalna do prędkości albo do jej kwadratu. Rozważając ruch obiektu w rzucie poziomy z uwzględnieniem oporu powietrza założymy, że siła oporu jest **wprost proporcjonalna** do prędkości. Wobec tego równania ruchu mają następującą postać:

$$ma_x = -kv_x \quad (1)$$

$$ma_y = -mg + kv_y \quad (2)$$

W tych równaniach **m** jest masą ciała,  $a_x$  oraz  $a_y$  oznaczają przyspieszenia,  $v_x$  oraz  $v_y$  oznaczają prędkości, **k** jest współczynnikiem oporu a **g** jest przyspieszeniem ziemskim.

W celu kompletnego opisu ruchu musimy określić warunki początkowe.

Zazwyczaj przyjmujemy, że ciało umieszczone jest na wysokości  $h$  metrów nad poziomem, w chwili początkowej  $t = 0$ , prędkość  $v_y$  wynosi zero, a prędkość  $v_x$  ma ustaloną wartość różną od zera. Wobec tego w chwili początkowej  $x = 0$  oraz  $y = h$ . Do modelowania ruchu wykorzystamy klasyczne techniki metod numerycznych, zastępując pochodne małymi przyrostami (przyspieszenie  $a$  jest równe pierwszej pochodnej prędkości  $v$  po czasie):

$$a = \frac{dv}{dt} \quad \Rightarrow \quad a = \frac{\Delta v}{\Delta t}$$

Do obliczenia przyspieszenia  $a_x$  wykorzystamy fakt, że przyrost prędkości  $\Delta v$  można obliczyć w kroku czasowym  $j+1$  znając wartość prędkości w kroku czasowym  $j$ :

$$a_x = \frac{\Delta v_x}{\Delta t} = \frac{v_x(t_{j+1}) - v_x(t_j)}{\Delta t}$$

Równanie ruchu w kierunku  $x$  w rzucie poziomym ma postać:

$$m \frac{v_x(t_{j+1}) - v_x(t_j)}{\Delta t} = -k v_x$$

co pozwala na wyznaczenie prędkości:

$$v_x(t_{j+1}) = v_x(t_j) - \frac{k}{m} v_x(t_j) \Delta t \quad (3)$$

Celem naszym jest wyznaczenie zależności przebytej drogi od czasu. Drogę  $\Delta x$  jaką pokona ciało w czasie  $\Delta t$  wyliczymy ze wzoru:

$$\Delta x = \frac{1}{2} (v_x(t_{j+1}) + v_x(t_j)) \Delta t$$

Korzystając z równania (3) mamy:

$$\Delta x = \frac{1}{2} \left( v_x(t_j) - \frac{k}{m} v_x(t_j) \Delta t + v_x(t_j) \right) \Delta t \quad (4)$$

Znając drogę w kroku czasowym  $j$  wyliczymy drogę w kroku czasowym  $j+1$ :

$$\begin{aligned} \Delta x &= x(t_{j+1}) - x(t_j) \\ x(t_{j+1}) &= \Delta x + x(t_j) \end{aligned}$$

Korzystając z równania (4) otrzymujemy wzór końcowy na zależność drogi w

kierunku  $x$  od czasu w rzucie poziomym z uwzględnieniem oporu powietrza:

$$x(t_{j+1}) = x(t_j) + v_x(t_j)\Delta t - \frac{k}{2m}v_x(t_j)(\Delta t)^2$$

W podobny sposób otrzymamy wzór na zależność drogi wzdłuż kierunku pionowym  $y$  od czasu:

$$y(t_{j+1}) = y(t_j) + v_y(t_j)\Delta t - \frac{1}{2}g(\Delta t)^2 + \frac{k}{2m}v_y(t_j)(\Delta t)^2$$

Program do wyznaczania trajektorii w rzucie poziomym pokazany jest na listingu 5.3.

### Listing 5.3. Modelowanie rzutu poziomego z oporem powietrza

---

```

#include <stdio.h>
#include <conio.h>
#include <math.h>

double simulate(double vx, double h, double m, double c, double dt, double eps);
int main()
{
    //    vo    h    m    c    dt    eps
    simulate( 55.0, 500.0, 20.0, 0.5, 0.005, 35.0);
    getch();
    return 0;
}

double simulate(double vx, double h, double m, double c, double dt, double eps)
{
    double v=0.0, vy=0.0, x=0.0, y=h, t=0.0;
    double r, g=9.81;
    r = c/m;
    printf("\n czas (s)  x(m)  y(m)  v(m/s)");
    int n = 0;
    while (y > -eps)
    {
        if ( !(n%100) )
            printf("\n %8.2f %8.2f %8.2f %8.2f", t,x,y,v);
        vx = vx - r*vx*dt;
        vy = vy - g*dt + r*vy*dt;
        x = x + vx*dt - 0.5*r*vx*dt*dt;
        y = y + vy*dt - 0.5*g*dt*dt + 0.5*r*vy*dt*dt;
        v = sqrt(vx*vx + vy*vy);
        t = t + dt;
        n++;
    }
    return t;
}

```

---

Wyniki obliczeń dla parametrów rzutu poziomego:

```
//      vo  h  m  c  dt  eps
simulate( 55.0, 500.0, 20.0, 0.5, 0.005, 35.0);
```

pokazane są w tabeli 5.4.

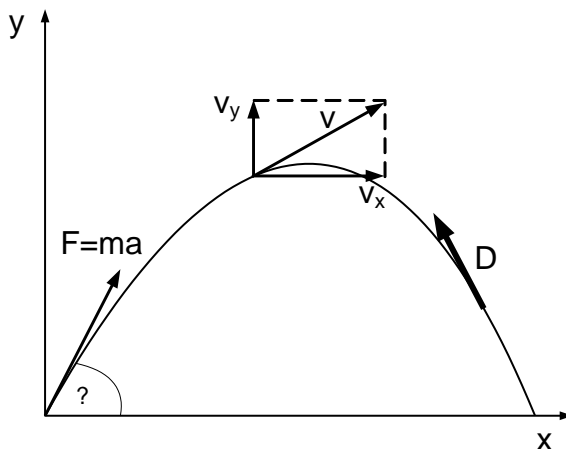
Tabela 5.4. Położenie oraz prędkość całkowita obiektu w rzucie poziomym z uwzględnieniem oporu powietrza w funkcji czasu

czas (s)	x (m)	y (m)	v (m/s)
0.00	0.00	500.00	0.00
0.50	27.33	498.74	54.54
1.00	54.31	495.00	54.55
1.50	80.96	488.75	55.06
2.00	107.28	479.95	56.05
2.50	133.27	468.57	57.53
3.00	158.94	454.58	59.48
3.50	184.29	437.95	61.86
4.00	209.33	418.64	64.65
4.50	234.06	396.61	67.81
5.00	258.47	371.85	71.31
5.50	282.59	344.30	75.12
6.00	306.40	313.94	79.20
6.50	329.92	280.74	83.54
7.00	353.15	244.64	88.11
7.50	376.09	205.62	92.88
8.00	398.74	163.65	97.85
8.50	421.11	118.68	103.00
9.00	443.21	70.68	108.31
9.50	465.03	19.60	113.78
10.00	486.57	-34.58	119.39

Modelujemy wyrzucenie kapsuły ratunkowej o masie 20 kilogramów z samolotu lecącego na wysokości 500 metrów nad poziomem morza, z prędkością 55m/s. Przyjęto (arbitralnie), że współczynnik oporu powietrza  $c$  dla kapsuły jest równy 0.5. Krok czasowy iteracji równy jest 0.005 sekundy. Wyświetlane na ekranie są wyniki z krokiem czasowym 0.5 sekundy (wyświetlany jest co setny wynik). Zwracamy uwagę, że tor już nie jest parabolą, tak jak w przypadku rzutu poziomego w próżni. Droga w kierunku poziomym znacznie się skróciła i wynosi około 470 metrów (w próżni ten zasięg byłby równy około 555 metrów).

#### 5.4. Rzut ukośny z uwzględnieniem oporu powietrza

W podręcznikach fizyki kinematyka i dynamika obiektów rozpatrywana jest w warunkach, gdy ruch nie jest zakłócany (np. ruch odbywa się w próżni). Dzięki takim założeniom, otrzymujemy przejrzyste formuły opisujące ruch. Niestety, gdy chcemy te wzory zastosować w praktyce, wyniki obliczeń znacznie odbiegają od danych realnych, co już pokazaliśmy. W rzeczywistości także opisując rzut ukośny musimy uwzględnić opór powietrza, gdyż wpływa on znacznie na tor poruszającego się obiektu.



Rys.5.2 Rzut ukośny, prędkość początkowa wynosi  $v_0$ , D - siła oporu

Zgodnie z drugą zasadą dynamiki Newtona równanie ruchu ma klasyczna postać:

$$F = m\vec{a}$$

Dla składowych  $x$  i  $y$  oraz przyspieszenia  $\mathbf{a}$  mamy układ równań:

$$ma_x = -kv_x$$

$$ma_y = -mg - kv_y$$

Przyspieszenie jest drugą pochodną położenia po czasie, wobec czego równania ruchu zależne od czasu mają postać:

$$m \frac{d^2x}{dt^2} = -k \frac{dx}{dt} \quad (1)$$

$$m \frac{d^2y}{dt^2} = -mg - k \frac{dy}{dt} \quad (2)$$

Są to równania liniowe rzędu drugiego. Warunki początkowe są oczywiste:

$$x(0) = 0, \quad y(0) = 0$$

Pamiętamy, że prędkość  $v_0$  jest wektorem:

$$\vec{v}_0 = (v_0 \cos \alpha, v_0 \sin \alpha)$$

Równania różniczkowe (1) i (2) przez podstawienie:

$$\frac{dx}{dt} = z$$

można sprowadzić do postaci:

$$\frac{dz}{dt} = -\frac{k}{m}z \quad (3)$$

Równanie (3) ma ogólne rozwiązanie analityczne postaci:

$$z(t) = C \exp\left(-\frac{k}{m}t\right) \quad (4)$$

Całkując to równanie i wykorzystując warunki początkowe dostajemy rozwiązanie  $x(t)$ . Postępując identycznie otrzymujemy rozwiązanie  $y(t)$ . Przy założeniu, że siła oporu w rzucie ukośnym jest proporcjonalna do prędkości początkowej otrzymujemy układ równań opisujących tor obiektu w rzucie ukośnym:

$$x(t) = \frac{mv_0}{k} \cos\alpha \left(1 - \exp\left(-\frac{k}{m}t\right)\right)$$

$$y(t) = \left(\frac{mv_0}{k} \sin\alpha + \frac{m^2g}{k^2}\right) \left(1 - \exp\left(-\frac{k}{m}t\right)\right) - \frac{mg}{k}t$$

Program pokazany na listingu 5.4 wylicza położenie obiektu (współrzędne  $x$  i  $y$ ) w funkcji czasu przy zadanych parametrach rzutu ukośnego z uwzględnieniem oporu powietrza. Współczynnik oporu  $k$  (w programie jest to stała  $c$ ) wybrany jest arbitralnie.

Listing 5.4. Modelowanie analityczne rzutu ukośnego z oporem powietrza

---

```
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define R 3.1415926/180.0

double simulate(double m, double v, double alfa, double c, double dt);
int main()
{
    // m v alfa c dt
    simulate(2.0, 20.0, 30.0, 0.70, 0.003);
    getch();
    return 0;
}
```

```
}  
  
double simulate(double m, double v, double alfa, double c, double dt)  
{ double x=0.0, y=0.0;  
  double t = 0.0;  
  double r,r1;  
  int n=0;  
  double g = 9.81;  
  r = c/m;  
  r1 = m*v/c;  
  double co = cos(R*alfa);  
  double si = sin(R*alfa);  
  printf("\n czas (s)  x(m)  y (m)");  
  do  
  {x=r1*co*(1-exp(-r*t));  
   y=(r1*si+g/(r*r))*(1-exp(-r*t))-g*t/r;  
   if (!(n%25) )  
     printf("\n %8.3f %8.2f %8.2f ",t,x,y);  
   t = t + dt;  
   n++;  
  } while (t <2.0);  
  return t;  
}
```

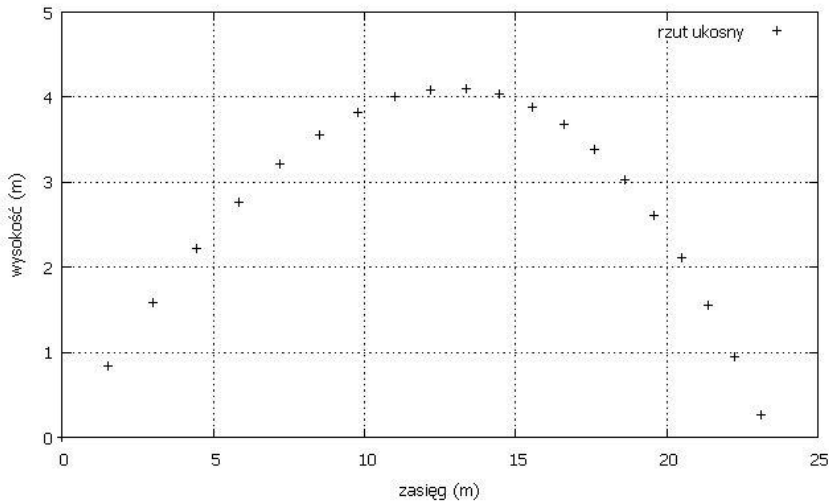
---

Program produkuje dużo danych, dlatego na ekranie monitora pokazywane są wybrane wyniki obliczeń (drukowany jest co 25 wynik) dzięki instrukcji:

```
if (!(n%25) )  
  printf("\n %8.3f %8.2f %8.2f ",t,x,y);
```

Wykres toru pokazany jest na rysunku 5.3





Rys. 5.3. Tor obiektu w rzucie ukośnym, parametry pokazane w listingu 5.4  
Dla parametrów rzutu:

```
//      m  v  alfa  c  dt
simulate( 2.0, 20.0, 30.0, 0.70, 0.003);
```

obiekt pokonuje odległość około 24 metrów w poziomie w czasie około 1.8 sekundy, wysokość najwyższego punktu toru wynosi około 4.14 metra.

Pokazany tor nie jest nie jest torem parabolicznym, jak to jest w rzucie ukośnym w próżni. Pokazana na rysunku 5.3 krzywa jest nazywana krzywą balistyczną.

Z punktu widzenia zastosowań w grafice komputerowej rozwiązanie analityczne pokazane powyżej może mieć jedną istotną wadę – jest nią koszt obliczeń. Obliczanie funkcji *exp()* wymaga dość długiego czasu. Wydaje się, że rozwiązania numeryczne zagadnienia rzutu ukośnego z uwzględnieniem oporu powietrza mogą być szybsze. Rozwiązując numerycznie równania ruchu dla przypadku rzutu ukośnego z uwzględnieniem oporu powietrza wykorzystamy stosowane już wcześniej przybliżenia:

$$a = \frac{dv}{dt} \quad \Rightarrow \quad a = \frac{\Delta v}{\Delta t}$$

Pokazane wcześniej równania ruchu (1) i (2) zostaną zastąpione wzorami:

$$m \frac{\Delta v_x}{\Delta t} = -k v_x$$

$$m \frac{\Delta v_y}{\Delta t} = -mg - k v_y$$

Rozwiązując pokazane równania ruchu zakładamy, że siła oporu powietrza jest wprost proporcjonalna do prędkości i przeciwnie skierowana do kierunku wektora prędkości. Obiekt zostaje wyrzucony pod kątem  $\alpha$  do poziomu z prędkością początkową  $v_0$ . W chwili początkowej  $t = 0$  warunki początkowe są następujące:

$$y(0) = 0, \quad x(0) = 0$$

oraz

$$v_x = v_0 \cos \alpha, \quad v_y = v_0 \sin \alpha$$

Wykorzystując poznane wcześniej techniki przybliżonego rozwiązywania równań różniczkowym otrzymujemy następujące wzory opisujące ruch obiektu w rzucie ukośnym z uwzględnieniem oporu powietrza.

Są to następujące równania:

$$v_x(t_{j+1}) = v_x(t_j) - \frac{k}{m} v_x(t_j) \Delta t$$

$$v_y(t_{j+1}) = v_y(t_j) - g \Delta t - \frac{k}{m} v_y(t_j) \Delta t$$

$$x(t_{j+1}) = x(t_j) + v_x(t_j) \Delta t - \frac{1}{2} \frac{k}{m} v_x(t_j) \Delta t^2$$

$$y(t_{j+1}) = y(t_j) + v_y(t_j) \Delta t - \frac{1}{2} g \Delta t^2 - \frac{1}{2} \frac{k}{m} v_y(t_j) \Delta t^2$$

Na listingu 5.5 przedstawiono program, dzięki któremu można modelować metodami numerycznymi rzut ukośny z uwzględnieniem oporu powietrza.

Listing 5.5. Modelowanie numeryczne rzutu ukośnego z oporem powietrza

```
//rozwiązanie numeryczne i analityczne, rzut ukośny z oporem
#include <stdio.h>
#include <conio.h>
#include <math.h>
#define PLIK "f:\\dysk d\\Dane\\wz.dat"
#define R 3.1415926/180.0
double simulate(double v,double alfa,double m,double c,double dt,double eps);

int main()
{
    //    vo  alfa  m  c  dt  eps
```

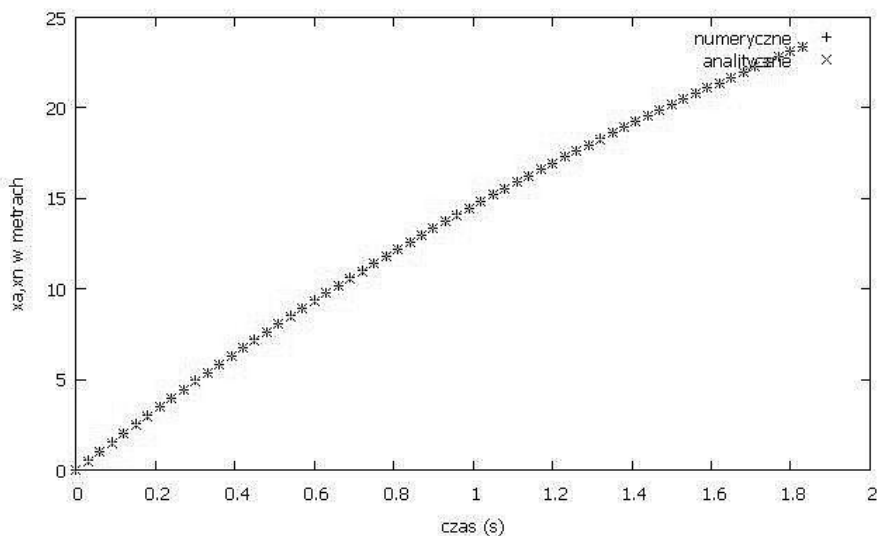
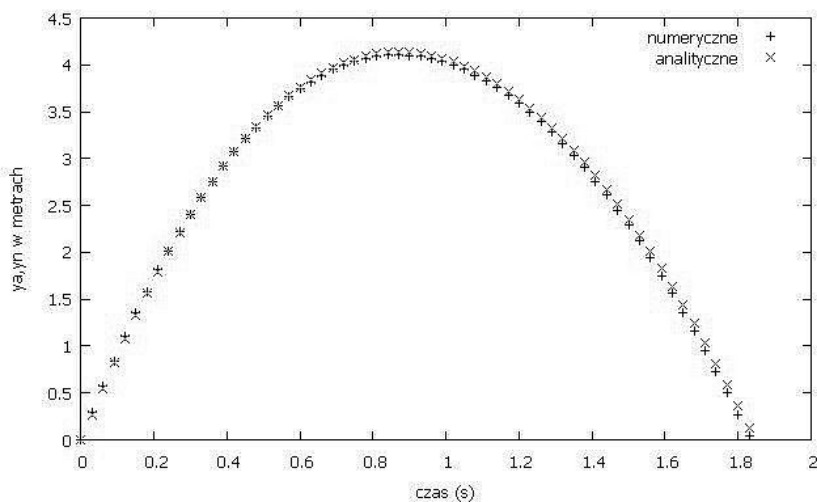
```

simulate( 20.0, 30.0, 2.0, 0.7, 0.003, 0.01);
getche();
return 0;
}

double simulate(double v,double alfa,double m,double c,double dt,double eps)
{ FILE *ws;
  double vx, vy, x=0.0, y=0.0, t=0.0 ;
  double xa=0.0, ya=0.0; //analityczne rozwiazanie
  double g=9.81;
  double r = c/m;
  double r1 = m*v/c;
  double co = cos(R*alfa);
  double si = sin(R*alfa);
  vx=v*co;
  vy=v*si;
  ws = fopen(PLIK, "w");
  printf("\n czas (s) x (m) y (m) ");
  int n = 0;
  while (y > -eps)
  { if ( !(n%10) )
    {
      printf("\n %8.2f %8.2f %8.2f ",t,x,y);
      fprintf(ws, "\n %8.3f %8.2f %8.2f %8.2f %8.2f",t,x,y,xa,ya);
    }
    // numeryczne -----
    vx = vx - r*vx*dt;
    vy = vy - g*dt - r*vy*dt;
    x = x + vx*dt - 0.5*r*vx*dt*dt;
    y = y + vy*dt - 0.5*g*dt*dt + 0.5*r*vy*dt*dt;
    // analityczne -----
    xa=r1*co*(1-exp(-r*t));
    ya=(r1*si+g/(r*r))*(1-exp(-r*t))-g*t/r ;
    // -----
    t = t + dt ;
    n++;
  }
  fclose(ws);
  return t;
}

```

W programie umieszczony jest także fragment kodu z programu obliczającego metodami analitycznymi tor obiektu w rzucie ukośnym w celu porównania wyników. Na rysunkach 5.4 i 5.5 przedstawiono porównanie wyników.

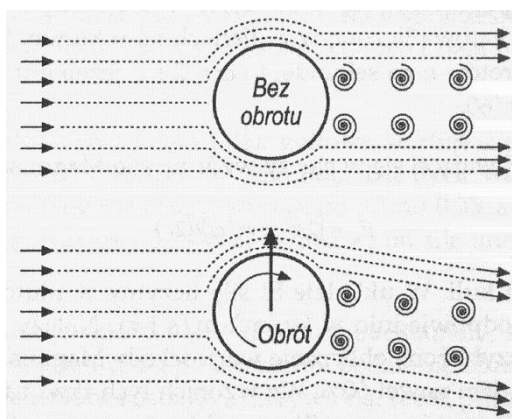
Rys.5.4 Zależność składowej  $x$  od czasu w rzucie ukośnymRys.5.5 Zależność składowej  $y$  od czasu w rzucie ukośnym

Analizując przedstawione wyniki, możemy uznać, że zgodność w wyznaczaniu toru obiektu w rzucie ukośnym metodami analitycznymi i metodami numerycznymi jest całkiem zadawalająca.

### 5.5. Efekt Magnusa i ruch obiektów

Obserwując ruch piłki golfowej, piłki baseballowej czy piłeczki pingpongowej zauważamy, że czasami wymienione obiekty poruszają się dziwnymi torami. Kibice sportowi z uznaniem mówią o graczach potrafiących „podkręcać” piłki. Z doświadczenia wiadomo, że obracające się ciało zachowuje się w ruchu inaczej niż poruszający się bez rotacji obiekt. Wiemy, że jeżeli ciało porusza się w powietrzu, doznaje działania siły oporu. Siła ta powoduje skrócenie (lub wydłużenie) zasięgu rzutu w porównaniu z zasięgiem ciała poruszającego się w próżni.

Jeżeli ciało porusza się w ośrodku (np. w powietrzu) i wiruje dookoła osi przechodzącej przez jej środek masy to pojawia się siła unoszenia. Pojawienie siły takiej siły nazywamy *efektem Magnusa*. Stosunkowo dawno zauważono wpływ rotacji na tor lotu obiektów, ale dopiero Henrich Magnus w 1853 roku dał teoretyczny opis tego zjawiska. Jeżeli powietrze opływa kulę (rys. 5.6), to za nią tworzy się strefa turbulencji (powstają wiry). Ciśnienie w strefie turbulencji jest mniejsze niż w strefie czołowej. Obrót kuli wokół osi przechodzącej przez jej środek masy powoduje przyspieszenie przepływu powietrza na górze i zmniejszenie przepływu na dole. Pojawia się siła unoszenia (dokładnie taka sama jak siła unoszenia na skrzydłach samolotu).



Rys. 5.6. Efekt Magnusa

Należy pamiętać, że siła Magnusa może być skierowana do góry lub do dołu, kierunek tej siły zależy od kierunku rotacji piłki w stosunku do kierunku ruchu. Pod działaniem siły Magnusa tor może zostać wydłużony lub skrócony. Siła Magnusa jest proporcjonalna do prędkości ciała, szybkości rotacji, gęstości ośrodka (powietrza) i rozmiarów ciała. Niestety, teoretycznie jest trudno wyliczyć siłę Magnusa – posługujemy się danymi doświadczalnymi lub wzorami przybliżonymi. Dla wirującej kuli przybliżony wzór na siłę Magnusa ma postać:

$$F_M = \rho v \omega \pi r^3$$

gdzie  $\rho$  jest gęstością powietrza ( $1.2 \text{ kg/m}^3$ ),  $\omega$  jest prędkością kątową liczoną w radianach na sekundę,  $v$  jest prędkością ciała,  $r$  jest promieniem kuli. Jeżeli określimy rotację, jako liczbę  $n$  obrotów na sekundę, to  $\omega = 2\pi n$ :

$$F_M = 2n\rho v\pi^2 r^3$$

Jak podaje w swojej monografii D. Bourg, tak obliczona siła może być obciążona błędem rzędu 50%. W praktyce posługujemy się tzw. współczynnikiem unoszenia  $C_L$  i wtedy wzór na siłę Magnusa ma postać:

$$F_M = (0.5\rho v^2 A)C_L$$

gdzie  $A$  jest powierzchnią przekroju poprzecznego.

Dla piłki golfowej, typowa prędkość to 76 m/s, piłka rotuje 30 - 60 razy na sekundę, kąt wyrzutu to około  $10^\circ$ . Współczynnik unoszenia zawiera się w przedziale od 0.1 do 0.35, maksymalna wartość to 0.45.

W przypadku piłki baseballowej typowa wartość prędkości to 45 m/s a piłka wykonuje około 30 obrotów na sekundę. Współczynnik unoszenia może mieć wartość od 0.1 do 0.2, maksymalna wartość to 0.45.

Równania ruchu z uwzględnieniem efektu Magnusa mają postać:

$$\begin{aligned} x &= v_x t \\ y &= v_y t - \left( g + \frac{F_M}{m} \right) \frac{t^2}{2} \end{aligned}$$

gdzie  $x, y$  to współrzędne obiektu,  $(v_x, v_y)$  to składowe prędkości początkowej  $v_0$ ,  $t$  jest czasem,  $m$  jest masą obiektu,  $F_M$  jest siłą Magnusa.

Po uwzględnieniu kąta wyrzutu  $\alpha$  mamy równania:

$$\begin{aligned} x &= (v_0 \cos \alpha) t \\ y &= (v_0 \sin \alpha) t - \left( g + \frac{F_M}{m} \right) \frac{t^2}{2} \end{aligned}$$

W rzucie ukośnym bez uwzględnienia oporów (rzut w próżni), współrzędna  $y$  ma postać:

$$y = (v_0 \sin \alpha) t - \frac{gt^2}{2}$$

Na listingu 5.6 przedstawiono program do modelowania ruchu piłki z uwzględnieniem efektu Magnusa.

## Listing 5.6. Modelowanie rzutu ukośnego z siłą Magnusa

```

#include <stdio.h>
#include <conio.h>
#include <math.h>

#define PLIK "f:\\dysk d\\Dane\\wm.dat"
#define Rd 3.1415926/180.0
#define PI 3.1415926

double simulate(double m, double R, double v, double alfa, int n1, double dt);
int main()
{
    //      m      R      v      alfa  n1  dt
    simulate(0.045, 0.02, 35.0, 20.0, 20, 0.01);
    getch();
    return 0;
}

double simulate(double m, double R, double v, double alfa, int n1, double dt)
{
    FILE *ws;
    double x=0.0, y=0.0, y1=0.0, y2=0.0;
    double t=0.0;
    int n=0;
    double g=9.81;
    double co=cos(Rd*alfa);
    double si=sin(Rd*alfa);
    double fm=2.0*n1*1.2*v*co*PI*PI*R*R*R;
    printf("\n Fm(N) = %6.4f v(m/s)= %5.2f n(obr.) = %3d \n", fm, v, n1);
    ws=fopen(PLIK, "w");
    printf("\n czas (s)  x(m)  y+ (m)  y- (m)  yp(m)");
    do
    {
        x=v*co*t;
        y=v*si*t-0.5*(g-fm/m)*t*t;
        y1=v*si*t-0.5*(g+fm/m)*t*t;
        y2=v*si*t-0.5*g*t*t;
        if (!(n%10))
        {
            printf("\n %8.3f %8.2f %8.2f %8.2f %8.2f", t, x, y, y1, y2);
            fprintf(ws, "\n %8.3f %8.2f %8.2f %8.2f %8.2f", t, x, y, y1, y2);
        }
        t=t+dt;
        n++;
    } while (t<3.5);
    fclose(ws);
    return t;
}

```

Należy pamiętać, że siła Magnusa może mieć wartość dodatnią lub ujemną w zależności od kierunku obrotu ciała w rzucie. Jeżeli w piłce (np. baseballowej czy golfowej) powierzchnia górna obraca się w kierunku swojego ruchu, to siła Magnusa będzie zakrzywiła tor piłki ku dołowi, co spowoduje krótszy zasięg w porównaniu z zasięgiem piłki bez rotacji.

W tabeli 5.5 pokazano wyniki modelowania toru obiektu w rzucie ukośnym z uwzględnieniem efektu Magnusa (siła Magnusa liczona jest dla kuli).

Parametry potrzebne do modelowania były następujące:

```
//      m  R  v  alfa  n  dt
simulate(0.045, 0.02, 35.0, 20.0, 20, 0.01);
```

Pokazane parametry to: m- masa piłki golfowej (kg), R – promień piłki (m), v – prędkość początkowa (m/s), alfa – kąt wyrzutu piłki, n1 – liczba obrotów na sekundę, dt – krok czasowy iteracji. Z tabeli 5.5. wynika, że w rzucie ukośnym zasięg piłki golfowej (przy prędkości początkowej 35 m/s i kącie wyrzutu  $20^\circ$ ) bez uwzględnienia oporu powietrza wynosi około 80 metrów (kolumna yp). W zależności od kierunku rotacji tor piłki wydłuży się lub skróci. Gdy piłka ma dodatnią siłę nośną (kolumna y+), to zasięg rzutu wynosi około 111 metrów, gdy siła nośna jest ujemna (kolumna y-) to zasięg jest skrócony do około 60 metrów. Te wyniki w zasadzie zgadzają się z danymi doświadczalnymi (pamiętamy, że dokładne wyliczenie siły Magnusa jest trudnym zadaniem). David Bourg w swojej monografii podaje, że przy kącie wyrzutu wzrost zasięgu spowodowany efektem Magnusa może wynieść nawet 60 metrów.

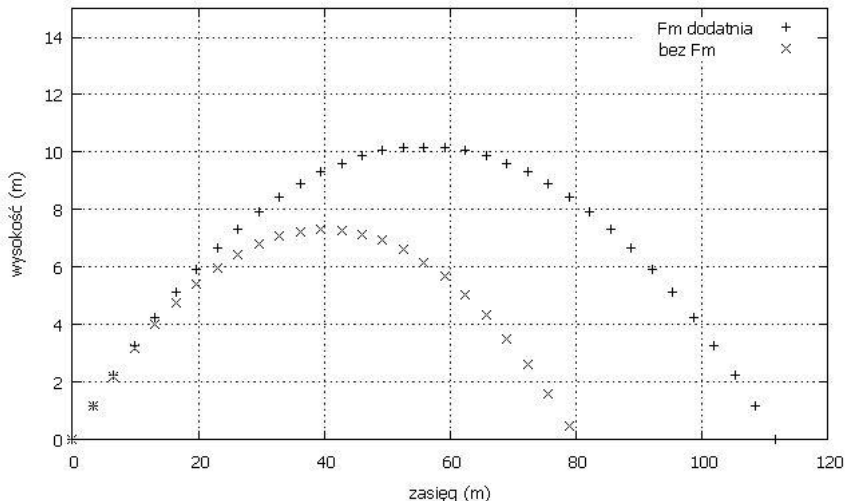
Tabela 5.5. Tor piłki golfowej z uwzględnieniem efektu Magnusa

Fm(N) = 0.1246    v(m/s)= 35.00    n(obr.) = 20				
czas (s)	x(m)	y+ (m)	y- (m)	yp(m)
0.000	0.00	0.00	0.00	0.00
0.200	6.58	2.25	2.14	2.20
0.400	13.16	4.23	3.78	4.00
0.600	19.73	5.92	4.92	5.42
0.800	26.31	7.32	5.55	6.44
1.000	32.89	8.45	5.68	7.07
1.200	39.47	9.30	5.31	7.30
1.400	46.04	9.86	4.43	7.15
1.600	52.62	10.14	3.05	6.60
1.800	59.20	10.14	1.17	5.66
2.000	65.78	9.86	-1.22	4.32
2.200	72.36	9.30	-4.11	2.60
2.400	78.93	8.45	-7.50	0.48
2.600	85.51	7.33	-11.40	-2.03
2.800	92.09	5.92	-15.80	-4.94
3.000	98.67	4.23	-20.70	-8.23
3.200	105.25	2.26	-26.10	-11.92
3.400	111.82	0.01	-32.01	-16.00
3.600	118.40	-2.52	-38.42	-20.47

Na rysunku 5.7 pokazano wykres toru piłki golfowej (parametry lotu podane

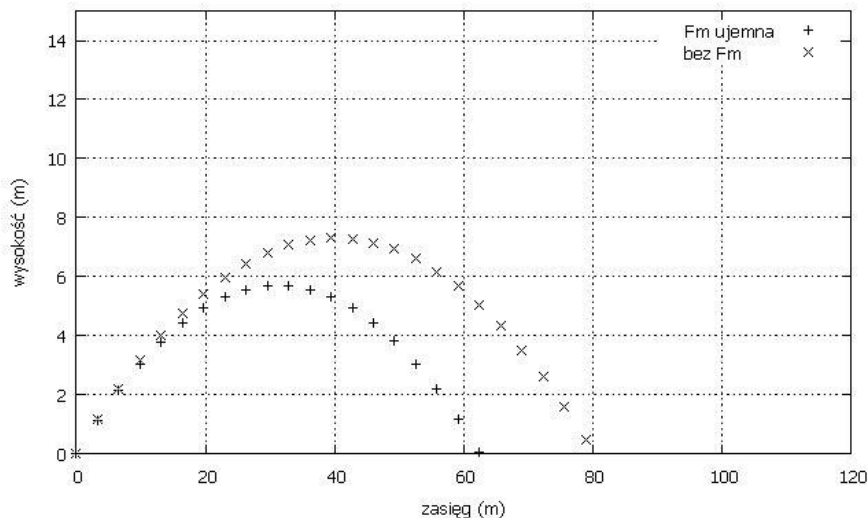


zostały, powyżej), gdy uwzględnimy efekt Magnusa w rzucie ukośnym. Pokazano tor piłki bez uwzględnienia oporu powietrza oraz z dodatnim efektem Magnusa (wydłużenie zasięgu). Zgodnie z obliczeniami zasięg bez uwzględnienia oporu powietrza wynosi około 80 metrów, z uwzględnieniem efektu Magnusa wzrasta do 110 metrów.



Rys. 5.7 Wykres toru piłki golfowej z dodatnim efektem Magnusa

Piłka może mieć rotację dodatnią lub ujemną. Zgodnie z umową, dodatnia wartość obrotu występuje wtedy, gdy punkt na dolnej części powierzchni kuli oddala się od gracza. Ujemny obrót występuje wtedy, gdy ten punkt przemieszcza się w kierunku gracza.



Rys. 5.8 Wykres toru piłki golfowej z ujemnym efektem Magnusa

Dodatni obrót wytwarza dodatnią siłę unoszenia i oczywiście powoduje zwiększenie zasięgu piłki. Ujemny obrót powoduje zmniejszenie zasięgu piłki. Na rys. 5.8. pokazano efekt ujemnej siły Magnusa. Zasięg piłki wynosi około 61 metrów i jest krótszy od zasięgu obliczonego dla lotu piłki bez efektu Magnusa (w naszym przypadku ten zasięg jest równy 80 metrów).

---

# ROZDZIAŁ 6

## MODELOWANIE RUCHU POJAZDÓW

---

6.1. Czynniki wpływające na ruch pojazdów.....	116
6.2. Kinetyka w opisie ruchu samochodów.....	117
6.3. Siły tarcia w ruchu samochodów .....	122
6.4. Siła dośrodkowa w ruchu samochodów .....	125

---

## 6.1. Czynniki wpływające na ruch pojazdów

Komputerowe symulacje ruchu pojazdów mają duże znaczenie ze względu na ich zastosowania praktyczne. Można wymienić wiele dziedzin, w których wymagamy modelowania ruchu pojazdów z różnym stopniem realizmu. W grach komputerowych wymagania związane z zastosowaniem precyzyjnych modeli fizycznych jest mniejsze niż w aplikacjach komputerowych tworzonych dla ekspertów sądowych odtwarzających przebieg wypadków samochodowych. Przykłady aplikacji, w których wykorzystujemy modelowanie ruchu samochodów to:

- gry komputerowe takie jak np. wyścigi samochodowe
- symulatory jazdy samochodem (szkoły dla kierowców)
- modelowanie wypadków samochodowych (firmy ubezpieczeniowe)
- modelowanie własności trakcyjnych (producenci samochodów)

Ta krótka lista nie wyczerpuje wszystkich zastosowań symulacji ruchu samochodów, ale pozwala nam zorientować się, że pola zastosowań są bardzo duże. Symulowanie ruchu samochodów można rozważać na różnym poziomie ogólności – od bardzo prostych modeli, takich jak ruch bryły sztywnej w przestrzeni jedno lub dwuwymiarowej, do skomplikowanych jak, na przykład bardzo techniczny opis ruchu samochodu formuły pierwszej (uwzględniając np. wpływ elementów aerodynamicznych, urządzenia typu KERS (ang. *Kinetic Energy Recovery System*), czy ruch po okręgu.

Symulując ruch samochodu należy posiadać minimum wiedzy z zakresu fizyki potrzebnych do poprawnego modelowania. Uważamy, że należy omówić takie zagadnienia jak:

- Równania ruchu 2D
- Siły oporu - powietrze
- Siły oporu - tarcie (statyczne i kinetyczne)
- Fizyka środka ciężkości
- Ruch po okręgu – tarcie
- Ruch po okręgu – przechył

W swojej monografii David Bourg wylicza sześć kroków, jakie należy wykonać modelując ruch samochodu:

- Obliczyć cechy charakterystyczne obiektu takie jak masa, moment bezwładności, środek masy,
- Wybrać istotne dla naszego modelu siły i momenty sił działające na obiekt,
- Obliczyć siły wypadkowe i momenty sił,
- Rozwiązać równanie ruchu (uwzględniając ruch postępowy i

- obrotowy),
- Scałkować równania ruchu po czasie w celu znalezienia prędkości liniowej i kątowej
- Ponownie scałkować równania ruchu po czasie w celu znalezienia przesunięcia liniowego i kątowego.

W kolejnych rozdziałach omówimy te zagadnienia.

## 6.2. Kinetyka w opisie ruchu samochodów

Wykorzystamy podstawowe elementy kinetyki (kinetyka zajmuje się opisem ruchu obiektów pod wpływem działających sił) do modelowania ruchu samochodów. W tym rozdziale przypomnimy niezbędne pojęcia kinetyki, potrzebne do opisu ruchu samochodu. Podstawowymi równaniami kinetyki są równania dynamiki Newtona.

Pierwsze prawo mówi, że gdy na obiekt nie działają siły (lub się równoważą) to prędkość obiektu pozostaje stała. Stąd wynikają dwa wnioski:

- jeżeli obiekt był w spoczynku, to nadal pozostanie w spoczynku,
- jeżeli obiekt porusza się w określonym kierunku z określoną prędkością, to nadal będzie poruszał się w tym samym kierunku z tą samą prędkością

Drugie prawo ruchu Newtona ma postać:

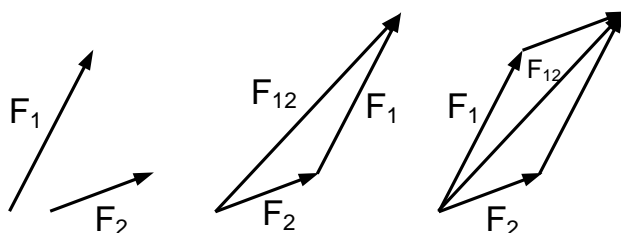
$$F = ma$$

Równanie to określa siłę, jako iloczyn masy i przyspieszenia. Interpretacja tego prawa jest następująca:

- jeżeli na obiekt działa siła (lub niezrównoważone siły) to następuje zmiana prędkości obiektu

Trzecie prawo Newtona mówi, że gdy dwa obiekty oddziałują ze sobą, siły, jakimi działają one na siebie mają taką samą wartość bezwzględną i przeciwne kierunki. Trzecie prawo Newtona wykorzystywane jest do opisu zderzeń samochodów. W grach komputerowych, programując kierowanie samochodem korzystamy z tych praw. Gdy samochód stoi a kierowca nie naciska pedału gazu, samochód dalej będzie stał (pierwsze prawo Newtona w praktyce). Gdy kierowca naciśnie pedał gazu, samochód zacznie przyspieszać, prędkość wzrasta (drugie prawo Newtona). Opisując ruch samochodu musimy uwzględnić siły na niego działające. Ponieważ siła jest wektorem, możemy wykorzystując reguły dodawania wektorów i obliczyć siłę wypadkową – jest to znaczna korzyść. Zamiast rozpatrywać każdą siłę z osobna i jej wpływ na ruch obiektu,

rozpatrujemy tylko wpływ siły wypadkowej na ruch. Na rysunku 6.1 pokazano sumowanie dwóch sił  $F_1$  i  $F_2$  działających na obiekt. Siła wypadkowa  $F_{12}$  została otrzymana dzięki metodzie geometrycznej dodawania wektorów. Wektor wypadkowy otrzymamy albo przesuając wektor  $F_1$  tak, aby jego początek był końcem wektora  $F_2$ , wtedy wektor wypadkowy  $F_{12}$  ma początek w początku wektora  $F_2$  a koniec w końcu wektora  $F_1$ , lub konstruując równoległobok i wtedy przekątna równoległoboku jest sumą wektorów tworzących jego boki.

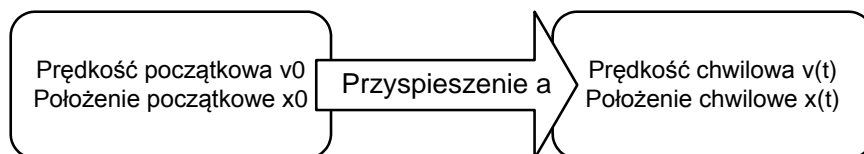


Rys.6.1 Dodawanie sił, na obiekt działają dwie siły:  $F_1$  i  $F_2$ . Wypadkowa siła to  $F_{12}$ . Na rysunku pokazano dwie techniki dodawania wektorów.

Na samochód w ruchu działać może wiele sił, najczęściej są:

- siła ciężkości,
- siła poruszająca samochód (silnik napędza koła),
- siła oporu powietrza,
- siła tarcia (tarcie opon o asfalt),
- siła dośrodkowa (samochód na zakręcie),

Modelując ruch samochodu musimy wyliczyć te wszystkie siły, znaleźć siłę wypadkową i obliczyć przyspieszenie (zmianę prędkości) spowodowane działaniem siły wypadkowej. Klasyczna mechanika jest deterministyczna, możemy przewidywać trajektorię (tor) obiektu w chwili późniejszej mając dane początkowe. Najprostsze dane początkowe to położenie początkowe  $x_0$ , prędkość początkowa  $v_0$  oraz działające siły, dzięki którym, znając masę obiektu możemy wyznaczyć przyspieszenie.



Rys. 6.2. Modelowanie parametrów ruchu w dowolnej chwili na podstawie warunków początkowych

W przypadku modelowania ruchu w przypadku jednowymiarowym, równania

ruchu wzdłuż linii prostej ze stałym przyspieszeniem mają postać:

$$v_{xn} = v_{xi} + a_x t \quad (\text{prędkość, jako funkcja czasu})$$

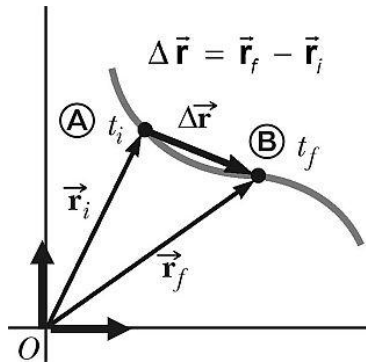
$$x_n = x_i + \frac{1}{2}(v_{xn} + v_{xi})t \quad (\text{położenie, jako funkcja prędkości i czasu})$$

$$x_n = x_i + v_{xi}t + \frac{1}{2}a_x t^2 \quad (\text{położenie, jako funkcja czasu})$$

$$v_{xn} = \sqrt{v_{xi}^2 + 2a_x(x_n - x_i)} \quad (\text{prędkość, jako funkcja czasu})$$

W tych wzorach  $v_{xn}$  i  $x_n$ , oznaczają wartości prędkości i położenia w kolejnych krokach czasowych,  $v_{xi}$  i  $x_i$  oznaczają wartości prędkości i położenie w poprzednich krokach,  $a_x$  jest przyspieszeniem,  $t$  oznacza czas.

Równania ruchu w jednym wymiarze możemy stosunkowo łatwo uogólnić na ruch w przestrzeni dwuwymiarowej. Praktycznie oznacza to modelowanie ruchu w jednej płaszczyźnie. Na rysunku 6.3 pokazano trajektorię obiektu przemieszczającego się z punktu A do punktu B. Położenie początkowe na płaszczyźnie w chwili  $t_i$  opisuje wektor położenia  $\vec{r}_i$ , położenie w punkcie B w chwili  $t_f$  opisuje wektor  $\vec{r}_f$ . Przemieszczenie charakteryzuje wektor  $\Delta\vec{r}$ .



Rys. 6.3. Trajektoria obiektu na płaszczyźnie

Wektor położenie na płaszczyźnie jest zdefiniowany następująco (i oraz j są wektorami jednostkowymi):

$$\vec{r} = x\hat{i} + y\hat{j}$$

Wektor prędkości na płaszczyźnie ma następującą postać:

$$\vec{v} = \frac{d\vec{r}}{dt} = \frac{dx}{dt}\hat{i} + \frac{dy}{dt}\hat{j} = v_x\hat{i} + v_y\hat{j}$$

przyspieszenie określamy wykorzystując prędkość:

$$\vec{a} \equiv \lim_{\Delta t \rightarrow 0} \frac{\Delta \vec{v}}{\Delta t} = \frac{d\vec{v}}{dt}$$

Gdy przyspieszenie jest stałe, równania ruchu są analogiczne do równań ruchu w przestrzeni jednowymiarowej:

$$\vec{v}_n = \vec{v}_i + \vec{a}t$$

$$\vec{r}_n = \vec{r}_i + \vec{v}_i t + \frac{1}{2} \vec{a} t^2$$

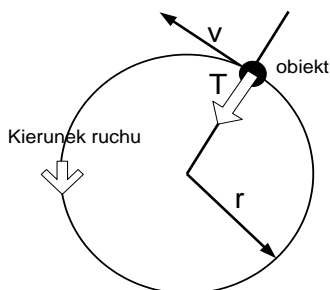
W tych wzorach  $v_n$  i  $r_n$  oznaczają prędkość i położenie w chwili  $t$ ,  $v_i$  oraz  $r_i$  oznaczają poprzednie prędkości i położenia. Powyższe równania sugerują, że ruch dwuwymiarowy możemy rozpatrywać, jako dwa niezależne ruchy: wzdłuż osi  $x$  oraz wzdłuż osi  $y$ . Gdy obiekt porusza się ze stałą prędkością  $v$  po okręgu (lub łuku) to taki ruch nosi nazwę ruchu jednostajnego po okręgu. Na obiekt w takim ruchu działa przyspieszenie dośrodkowe  $a$  o stałej wartości:

$$a = \frac{v^2}{r}$$

W tym wzorze  $r$  jest promieniem okręgu. Przyspieszenie jest skierowane do środka okręgu. Wiemy, że źródłem przyspieszenia jest siła dośrodkowa  $T$ , która jest oczywiście też skierowana do środka okręgu:

$$T = ma = m \frac{v^2}{r}$$

W ruchu jednostajnym po okręgu wektor prędkości  $v$  jest zawsze stycznie skierowany do toru. Siła dośrodkowa nadaje obiektowi przyspieszenia, zmieniając kierunek prędkości, ale nie zmienia wartości tej prędkości.

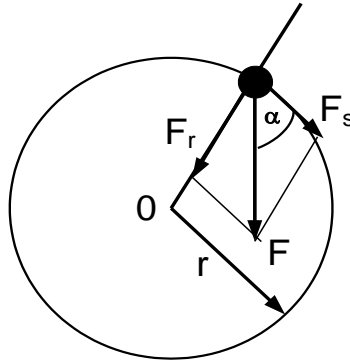


Rys. 6.4. Siła dośrodkowa w ruchu po okręgu.

Rozpatrując ruch obrotowy bryły sztywnej wprowadza się pojęcie momentów



sił. Rozpatrzmy ruch pojedynczego punktu materialnego o masie  $m$  poruszającego się po okręgu o promieniu  $r$ .



Rys. 6.5. Obrót punktu po okręgu – moment siły

Niech na punkt działa siła  $F$ . Punkt materialny nabywa stałego przyspieszenia stycznego  $a_s$  spowodowanego przez składową siły  $F_s$ :

$$F_s = F \cos \alpha = ma_s$$

Składowa normalną siły  $F$  równoważy reakcja więzów. Jeżeli wprowadzimy przyspieszenie kątowe  $\varepsilon$ :

$$\varepsilon = \frac{a_s}{r}$$

to równanie przybierze postać:

$$F \cos \alpha = mr\varepsilon$$

Możemy obie strony równania pomnożyć przez  $r$ :

$$Fr \cos \alpha = mr^2\varepsilon$$

Wielkość

$$M = Fr \cos \alpha$$

nosi nazwę **momentu siły** względem punktu  $O$ .

Wielkość

$$I = mr^2$$

nazywa się momentem bezwładności punktu materialnego względem punktu  $O$ . Moment siły i moment bezwładności związany jest równaniem:

$$M = I\varepsilon$$

Jest to zasada dynamiki Newtona ( $F = ma$ ) wyrażona dla ruchu obrotowego. Formalnie moment siły jest iloczynem wektorowym:

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

### 6.3. Siły tarcia w ruchu samochodów

Różne czynniki mogą wpływać na ruch samochodów. Do najważniejszych zaliczamy siły takie jak siła tarcia oraz siła oporu powietrza.

W życiu codziennym widzimy, że obiekt w ruchu po jakimś czasie zatrzyma się (mimo obowiązującego pierwszego prawa Newtona). Musiały działać jakieś siły skierowane w przeciwnym kierunku do siły wprawiającej obiekt w ruch. Najczęściej są to siły tarcia. Rozróżniamy siły tarcia statycznego (są to siły działające, gdy ciało jest w spoczynku) oraz siły tarcia kinetycznego (występuje podczas ruchu obiektu). Wartość siły tarcia kinetycznego  $F_k$  jest zazwyczaj mniejsza od siły tarcia statycznego  $F_s$ . Maksymalne wartości siły tarcia dane są wzorami:

$$F_s = f_s N \quad (1)$$

$$F_k = f_k N \quad (2)$$

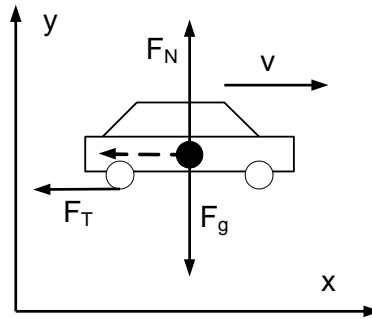
W tych wzorach  $f_s$  i  $f_k$  oznaczają współczynniki tarcia,  $N$  jest siłą normalną (jest to siła działająca na obiekt stojący na poziomym podłożu spowodowana grawitacją,  $N = mg$ ). Współczynniki tarcia są wyznaczone doświadczalnie, zależą one od rodzaju materiałów kontaktujących się. Zaskakującym faktem, może być obserwacja, że tarcie statyczne nie zależy do wielkości kontaktujących się powierzchni. W tabeli 6.1. Pokazano współczynniki tarcia.

Tabela 6.1. Wybrane współczynniki tarcia

Material	$f_s$	$f_k$
Opona na suchym betonie	1.0	0.6 – 0.8
Opona na mokrym betonie	0.7	0.5
Opona na oblodzonym betonie	0.3	0.02
Guma na suchym asfalcie	0.6	0.4
Drzewo po drzewie	0.5	0.3
Stal po stali	0.6	0.4
Stal po stali z olejem	0.1	0.05
Teflon po stali	0.04	0.04

Hamując samochód łatwo można doprowadzić do zablokowania kół – mamy do

czynienia z tzw. poślizgiem. Należy pamiętać, że w praktyce tarcie kinetyczne maleje, gdy prędkość poślizgu wzrasta. Zmierzono, że zablokowana opona samochodu ma współczynnik tarcia około 0.8 przy prędkości 8 km/h, natomiast przy prędkości 130 km/h ten współczynnik może mieć wartość mniejszą niż 0.5. Dlatego producenci samochodów wprowadzają zabezpieczenia przeciwdziałające blokowaniu kół.



Rys.6.6 Siły działające na samochód jadący prosto.  $F_N$  – siła normalna,  $F_g$  - siła ciężkości,  $F_T$  - siła tarcia. Pomijamy opór powietrza.

Bardzo często podczas hamowania awaryjnego koła zostają zablokowane. Na asfalcie lub betonie pozostaje ślad ślizgających się opon. Dzięki pomiarowi długości śladu hamowania możemy z bardzo małym błędem określić prędkość samochodu w momencie zablokowania kół. Jeżeli założymy, że przyspieszenie (w zasadzie mówimy o opóźnieniu) było stałe to równanie ruchu ma postać:

$$v = \sqrt{v_0^2 + 2a(x - x_0)}$$

gdzie  $v_0$  jest to prędkość początkowa, przemieszczenie wynosi  $(x-x_0)$ ,  $a$  jest przyspieszeniem,  $v$  jest prędkością. Ponieważ samochód się zatrzymał, to  $v$  jest równe zero. Nie znamy jedynie przyspieszenia. Przyspieszenie  $a$  jest wywołane siłą tarcia ( $m$  jest masą samochodu):

$$-F_t = ma$$

Znak minus pojawia się w równaniu, dlatego, że siła tarcia kinetycznego jest skierowana przeciwnie do kierunku ruchu samochodu. Wiemy także, że siła tarcia wyraża się wzorem (2), ponieważ mamy do czynienia z tarcie kinetycznym, gdzie  $N$  jest wartością siły normalnej a  $f_k$  jest współczynnikiem tarcia kinetycznego. Ponieważ  $N = mg$ , to mamy:

$$a = -\frac{F_t}{m} = -\frac{f_k mg}{m} = -f_k g$$

Tak wyliczone przyspieszenie wykorzystujemy we wzorze na prędkość. Ponieważ  $v = 0$ , to otrzymujemy wzór:

$$\sqrt{v_0^2 + 2a(x - x_0)} = 0$$

Prędkość początkowa dana jest wzorem:

$$v_0 = \sqrt{2f_k g(x - x_0)}$$

Jeżeli długość śladu wynosiła na przykład 300 m, to przyjmując, że  $f_k = 0.6$ , prędkość początkowa samochodu  $v_0$  w tym przypadku wynosiła:

$$v_0 = \sqrt{(2)(0.6)(9.81)(300)} = 59.42 \left(\frac{m}{s}\right)$$

Prędkość początkowa, w chwili rozpoczęcia hamowania awaryjnego w tym przypadku wynosiła około 59 m/s, co wynosi około 214 km/h! Ten wynik wyjaśnia, dlaczego na autostradach dochodzi do tzw. masowych karamboli, podczas których wpada na siebie czasem ponad 50 samochodów. W czasie mgły, widoczność zmniejsza się, a w momencie zablokowania kół, kierowca nie może już nic uczynić.

Kierowca na drodze musi często podejmować decyzję, aby uniknąć wypadku. Jako przykład rozpatrzmy następującą sytuację: kierowca jadący drogą zobaczył przed sobą nieruchomą przeszkodę. Czy w takiej sytuacji powinien hamować, czy próbować zakręcić? Odpowiedź nie jest oczywista, musimy wykonać odpowiednie obliczenia.

Jeżeli kierowca hamuje, siła tarcia

$$F_T = f_s mg$$

powoduje opóźnienie  $a$ :

$$a = \frac{F_T}{m} = f_s g$$

W tym przypadku samochód przebędzie drogę  $h$ :

$$h = \frac{at^2}{2}$$

W tym wzorze  $t$  oznacza czas hamowania. Ponieważ

$$t = \frac{v_0}{a}$$

to droga hamowania wyraża się wzorem:

$$h = \frac{v_0^2}{2f_s g}$$

Rozpatrzmy drugą opcję – kierowca próbuje zakręcić. W takiej sytuacji najmniejszy promień skrętu, przy którym kierowca nie wprowadzi samochodu w poślizg zależy od siły tarcia  $f_s$ , dzięki której samochód uzyskuje przyspieszenie dośrodkowe  $a$ :

$$a = \frac{v_0^2}{r}$$

Z drugiego prawa dynamiki Newtona mamy równość:

$$m g f_s = m \frac{v_0^2}{r}$$

Z tej równości możemy obliczyć promień skrętu:

$$r = \frac{v_0^2}{f_s g}$$

Widzimy, że droga hamowania jest dwa razy mniejsza niż promień skrętu, co oznacza, że w opisanej sytuacji należy hamować a nie próbować omijać przeszkodę!

#### 6.4. Siła dośrodkowa w ruchu samochodów

Na samochód pokonujący zakręt (rys. 6.7) działa siła dośrodkowa. Gdy samochód jedzie po łuku, porusza się ruchem jednostajnym po okręgu, doznaje przyspieszenia skierowanego do środka łuku. Ta siła dośrodkowa jest siłą tarcia, jaką droga (np. asfalt) działa na opony kół samochodu. Tylko dzięki sile tarcia samochód ma możliwość pokonania zakrętu. Istnieje prędkość graniczna (maksymalna prędkość), z jaką można pokonywać zakręty. Po przekroczeniu określonej prędkości samochód traci przyczepność i „wypada” z trasy. Określmy tę maksymalną prędkość.

Przyjmijmy następujące warunki: samochód ma masę równą 1500 kg, niech promień zakrętu jest równy 35 metrów, przyjmijmy współczynnik tarcia  $f_s$  równy 0.5.

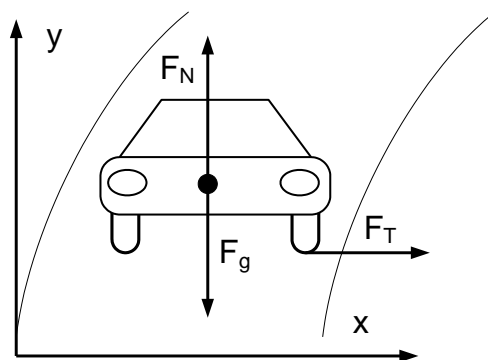
Naszym zadaniem jest wyliczenie maksymalnej prędkości, z jaką można w tych

warunkach pokonać zakręt.



Rys. 6.7. Samochód pokonuje zakręt, działa siła dośrodkowa

Wiemy, że siłą dośrodkową jest siła tarcia  $F_T$ . Ponieważ samochód nie ślizga się, jest to siła tarcia statycznego. W warunkach maksymalnej prędkości, siła dośrodkowa jest równoważona siłą tarcia. Gdyby siła tarcia miała wartość zerową (na przykład samochód porusza się po lodzie), samochód jechałby prosto, to znaczy po chwili wypadałby z drogi.



Rys. 6.8. Samochód pokonuje zakręt, działające siły,  $F_T$  – siła tarcia

W warunkach równowagi druga zasada dynamiki Newtona wiąże siłę tarcia i siłę dośrodkową dla składowych działających wzdłuż osi  $r$  :

$$F_T = m \frac{v^2}{r}$$

Pamiętając, że siła normalna  $F_N = mg$ , mamy wzór na siłę tarcia:

$$F_T = f_s mg$$

Wobec tego równowaga sił ma postać:

$$f_s mg = m \frac{v^2}{r}$$

Z tego równania możemy otrzymać wzór na prędkość maksymalną podczas ruchu samochodu po łuku:

$$v = \sqrt{f_s gr}$$

Prędkość maksymalna w takim ruchu nie zależy od masy pojazdu!

Wylczenie prędkości maksymalnej dla rozpatrywanego przypadku jest już proste:

$$v = \sqrt{f_s gr} = \sqrt{(0.5) \left( \frac{9.81m}{s^2} \right) (35.0m)} = 13.1 \left( \frac{m}{s} \right)$$

Widzimy, że na łuku o tak małym promieniu ( $r = 35 \text{ m}$ ), gdy współczynnik tarcia statycznego  $f_s = 0.5$ , prędkość maksymalna tuż przed utratą przyczepności wynosi około 47.7 km/h.

Współczynnik tarcia statycznego  $f_s$  odgrywa dużą rolę w bezpiecznym ruchu samochodów. Możemy oszacować ten współczynnik z następującej obserwacji. W deszczowy dzień, rozważany samochód pokonuje łuk o promieniu 35 metrów. Obserwujemy, że przy prędkości około 30 km/h samochód traci przyczepność. Możemy oszacować  $f_s$  ze wzoru:

$$f_s = \frac{v^2}{gr} = \frac{\left( \frac{8.33m}{s} \right)^2}{\left( \frac{9.81m}{s^2} \right) (35.0m)} = 0.20$$

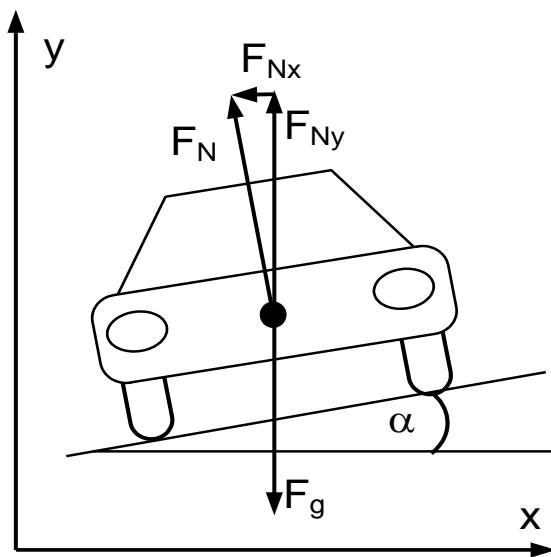
Widzimy, jak silnie może zmienić się współczynnik tarcia statycznego na mokrej nawierzchni. W powyższych obliczeniach prędkość 30 km/h to w przybliżeniu 8.33 m/s.

Możemy znacznie zwiększyć maksymalną prędkość samochodu na łuku drogi, jeżeli jezdnia jest odpowiednio wyprofilowana (rys. 6.9). Przy odpowiednim nachyleniu powierzchni drogi, siły tarcia nie odgrywają dominującej roli.

Przy odpowiednim profilu jezdni może być oblodzona (bardzo mały

współczynnik tarcia) a mimo to samochód może jechać ze znaczną prędkością. Obliczymy, jakie musi być nachylenie powierzchni jezdni, aby samochód poruszający się po łuku drogi o promieniu 200 metrów mógł przejechać z prędkością 100 km/h. Przyjmijmy, że współczynnik tarcia statycznego jest równy zero. Jak wynika z rys. 6.9., gdy jezdnia jest nachylona to pojawia się składowa pozioma siły normalnej ( $F_{Ny} = F_N \sin \alpha$ ).

Ta składowa jest odpowiedzialna za przyspieszenie dośrodkowe.



Rys. 6.9. Samochód na drodze wyprofilowanej

W tej sytuacji drugie prawo dynamiki Newtona dla składowej radialnej siły (kierunek  $x$ ) ma postać:

$$F_N \sin \alpha = \frac{mv^2}{r}$$

Drugie prawo dynamiki Newtona dla składowej pionowej siły (kierunek  $y$ ) ma postać:

$$F_N \cos \alpha = mg$$

Dzieląc składowe poziome siły przez składowe pionowe siły otrzymujemy następujący wzór:

$$\tan \alpha = \frac{v^2}{rg}$$

Biorąc pod uwagę nasze dane (prędkość  $v = 100$  km/h, promień łuku  $r = 200$  m)

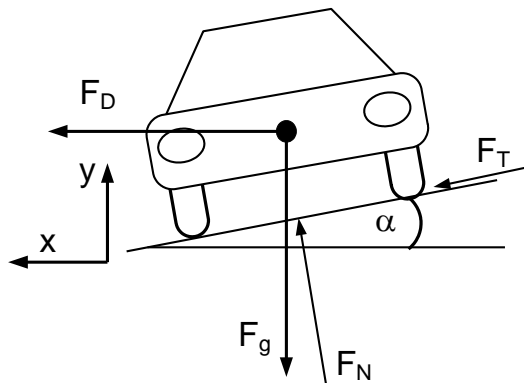


możemy oszacować kąt nachylenia:

$$\alpha = \tan^{-1} \left( \frac{\left( \frac{27.77 \text{ m}}{\text{s}} \right)^2}{\left( \frac{200 \text{ m}}{\text{s}} \right) \left( \frac{9.81 \text{ m}}{\text{s}^2} \right)} \right) = 0.393 \text{ rd}$$

Wynik podany jest w radianach, po przeliczeniu na stopnie, widzimy, że jezdnia powinna być pochylona pod kątem 21.5 stopnia, aby bez względu na warunki pogodowe samochód mógł pokonać łuk drogi z prędkością 100 km/h. W pokazanym wzorze muszą być jednakowe jednostki, dlatego prędkość wyrażona w km/h została przeliczona do prędkości podanej w m/s (100 km/h = 27.77 m/s). Oczywiście tarcie jest wymagane, aby samochód nie ześlizgiwał się z drogi (ruch po równi pochyłej). Otrzymany wynik ponownie nie zależy od masy pojazdu. Co ciekawsze, gdy występuje niezerowe tarcie statyczne, samochód może pokonać łuk drogi znacznie szybciej. Maksymalna prędkość zależy, więc od kąta nachylenia jezdni oraz od współczynnika tarcia statycznego.

Rozważymy ponownie ruch samochodu po łuku jezdni nachylonej pod kątem  $\alpha$ , ale teraz uwzględnimy występowanie siły tarcia (rys. 6.10)



Rys. 6.10. Ruch samochodu po łuku jezdni nachylonej pod kątem  $\alpha$

Jeżeli oznaczymy przez  $v$  maksymalną prędkość, przy której samochód nie straci przyczepności oraz przyjmiemy, że współczynnik tarcia statycznego jest równy  $f_s$  to możemy zapisać równania działających sił.

Rozważając składowe działających sił w kierunku osi  $x$  widzimy, że siła dośrodkowa równoważona jest przez składową siły normalnej i siłą tarcia:

$$m \frac{v^2}{r} = N \sin \alpha + f_s N \cos \alpha$$

Podobnie dla składowych sił działających w kierunku osi  $y$  ( $F_g = mg$ ) możemy

napisać:

$$N \cos \alpha - f_s N \sin \alpha - mg = 0$$

Przekształcając powyższe równania otrzymujemy:

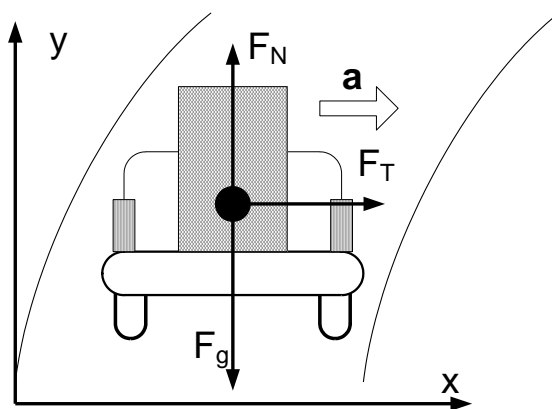
$$m \frac{v^2}{r} = N(\sin \alpha + f_s \cos \alpha)$$

$$mg = N(\cos \alpha - f_s \sin \alpha)$$

Dzieląc równanie pierwsze przez drugie, otrzymujemy wzór na maksymalną prędkość samochodu pokonującego łuk drogi pochylonej pod kątem  $\alpha$ , gdy występuje tarcie  $f_s$ :

$$v = \sqrt{\frac{rg(\sin \alpha + f_s \cos \alpha)}{(\cos \alpha - f_s \sin \alpha)}}$$

Często zdarza się, że ciężarówka przewozi kontener, który postawiony jest na jej podłodze. Kierowcy często sądzą, że ciężkiego kontenera nic nie jest w stanie ruszyć, wobec czego nie zabezpieczają odpowiednio ładunku. Okazuje się jednak, że przy dużej prędkości, na łuku drogi może nastąpić przesunięcie ładunku. Naszym zadaniem będzie wyliczenie maksymalnej prędkości ciężarówki poruszającej się po łuku płaskiej drogi o określonym promieniu, tak, aby nie doszło do ruchu kontenera po podłodze platformy ciężarówki. Przyjmiemy, że współczynniki tarcia statycznego pomiędzy kontenerem i podłogą wynosi 0.6, a promień łuku jest równy 35 metrów.



Rys. 6.11. Samochód na łuku drogi z przewożonym ładunkiem  
Ponieważ przyspieszenie pionowe  $a_y = 0$ , to siła  $F_N$  wynosi:

$$F_N = mg$$

Siłą powodującą przyspieszenie dośrodkowe  $a_r$  jest siła tarcia, z drugiej zasady dynamiki Newtona mamy:

$$F_T = ma_r = \frac{mv^2}{r}$$

Ponieważ musi być spełniony warunek:

$$F_T \leq f_s F_N$$

to z nierówności:

$$\frac{mv^2}{r} \leq f_s mg$$

Otrzymujemy wzór na maksymalną prędkość:

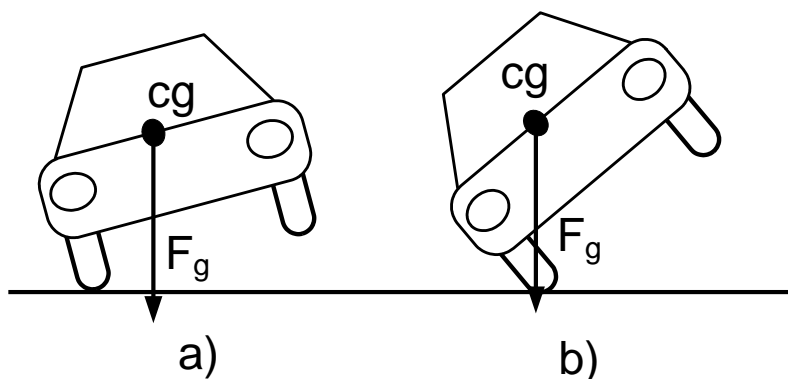
$$v = \sqrt{f_s r g}$$

Po wstawieniu odpowiednich wartości:

$$v \leq \sqrt{0.6 \left(35 \frac{m}{s}\right) \left(9.81 \frac{m}{s^2}\right)} = 14.3 \frac{m}{s}$$

Jeżeli samochód nie przekroczy prędkości 51.5 km/h (14.3 m/s) to kontener nie przesunie się.

Położenie środka ciężkości samochodu ( $cg$  – ang. centre of gravity) ma wpływ na jego stabilność. Im niżej jest położony środek ciężkości tym stabilniej na drodze zachowuje się samochód. Jeżeli położenie środka ciężkości jest wysokie (np. w ciężarówkach) to większe jest prawdopodobieństwo przewrócenia się samochodu, gdy pojawi się jakaś siła boczna (np. wiatr lub siła dośrodkowa).



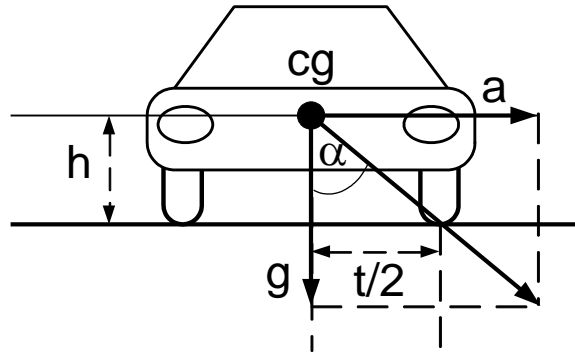
Rys. 6.12. Stabilność samochodu

Samochody wyścigowe mają bardzo nisko położony środek ciężkości i są szerokie, co pozwala im pokonywać zakręty o małym promieniu z dużą szybkością. Samochód przewróci się pod działaniem siły bocznej, gdy kierunek siły ciężkości wyprowadzony ze środka ciężkości znajdzie się poza punktem jego podparcia.

Zagadnienie stabilności samochodów jest ważne, Ministerstwo Transportu w USA (*The National Highway Traffic Safety Administration (NHTSA) of the Department of Transportation*) około 2000 roku opublikowało dane dotyczące podatności różnych samochodów na przewrócenie się. Parametr stabilności **SSF** (ang. *Static Stability Factor*) został zmierzony dla wielu popularnych samochodów. Ten parametr jest zdefiniowany, jako iloraz połowy szerokości pojazdu i wysokości środka ciężkości (rys. 6.13):

$$SSF = \frac{0.5t}{h}$$

Rozpatrzmy myślowy eksperyment opisujący przewracanie się samochodu. Samochód z określoną prędkością (pod działaniem siły bocznej) przesuwa się w prawo aż do momentu, gdy prawa opona napotka na przeszkodę (np. krawężnik). Środek tej prawej opony jest punktem obrotu, samochód nie może się dalej przesuwać, działa siła boczna, dochodzi do wywrócenia pojazdu. Wywrócenie zachodzi wtedy, gdy wektor wypadkowy siły bocznej i siły ciężkości przechodzi przez środek prawej opony (rys. 6.13.).



Rys. 6.13 Składowe SSF dla krytycznej siły bocznej

Samochód zacznie się obracać wokół punktu obrotu, gdy zostanie spełniony warunek:

$$\tan \alpha = \frac{a}{g} = \frac{0.5t}{h} = \frac{F_l}{F_g}$$

Oznaczenia są następujące:  $g$  - przyspieszenie ziemskie,  $a$  – przyspieszenie wywołane siłą boczną,  $F_l$  – siła boczna,  $F_g$  – siła ciężkości.

Można wykazać, że krytyczna prędkość  $v$ , przy której nastąpi wywrócenie samochodu w opisanej sytuacji wyraża się wzorem:

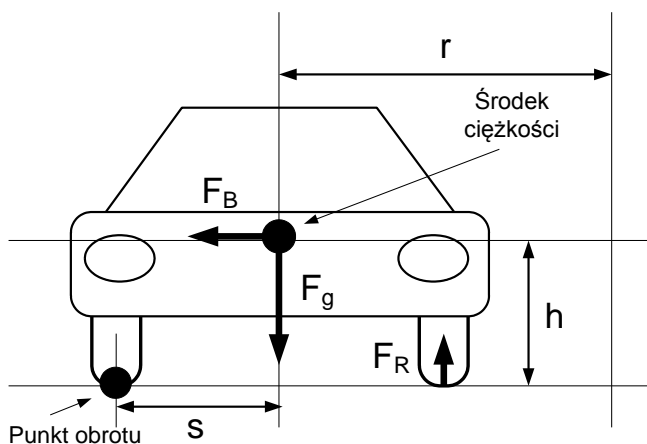
$$v^2 = 2g \sqrt{\left(\frac{t}{2}\right)^2 + h^2} - h$$

Według badań NHTS parametr stabilności spełnia warunek:

$$1.0 \leq SSF \leq 1.5$$

Zgodnie z logiką żądamy, aby prędkość krytyczna  $v$  była jak największa, co oznacza, że  $h$  powinno być małe (środek ciężkości powinien leżeć tak nisko jak to jest możliwe) oraz rozstaw kół  $t$  powinien być tak szeroki jak to jest możliwe. Jeżeli samochód ma wysoko umieszczony środek ciężkości to może się zdarzyć, że pokonując zakręt o zbyt małym promieniu, mając zbyt dużą prędkość może się przewrócić. Przyjmiemy, że środek ciężkości znajduje się na wysokości  $h$  a rozstaw kół wynosi  $2s$ . Wyznamy prędkość krytyczną  $v$ . Na samochód działa siła ciężkości  $F_g$  i siła bezwładności  $F_B$ . Jezdnia działa na lewe koło z siłą  $F_l$ , a na prawe z siłą  $F_r$ .

Jeżeli moment siły  $F_B$  względem lewego koła jest większy od momentu siły  $mg$  względem tego koła, to samochód zostanie obrócony wokół punktu styku tego koła i jezdni i wywróci się.



Rys. 6.14. Siły powodujące wywrócenie się samochodu

W warunkach równowagi znika suma momentów względem tego punktu podparcia (rys. 6.14):

$$F_B h + mgs + F_R 2s = 0$$

Przekształcając to równanie otrzymujemy:

$$F_R = \frac{mg}{2} - \frac{F_B h}{2s} \geq 0$$

Ponieważ

$$F_B = \frac{mv^2}{r}$$

oraz

$$F_R = -mg$$

to ostatecznie otrzymujemy wzór na prędkość krytyczną:

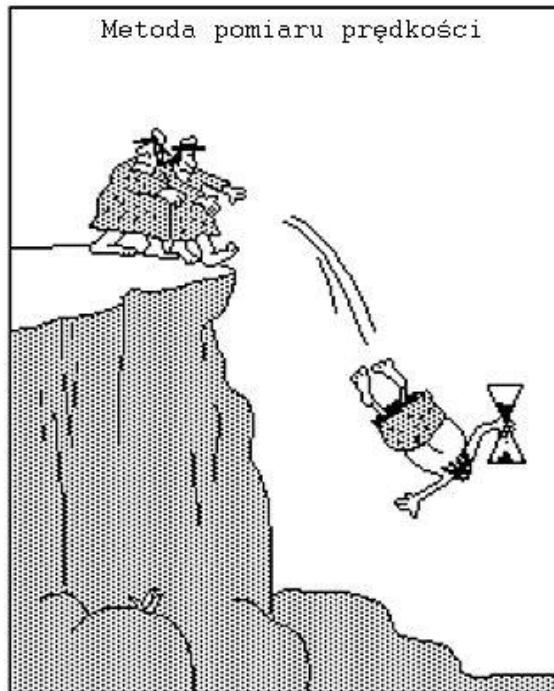
$$v \leq \sqrt{\frac{rsg}{h}}$$

Na podstawie otrzymanego wzoru możemy oszacować prędkość krytyczną, prowadzącą do przewrócenia się samochodu. Niech samochód dostawczy ma rozstaw kół równy 1.5 metra, środek ciężkości niech będzie na wysokości 2 metrów (samochód przewozi wysoki ładunek) a promień skrętu niech będzie równy 5 metrów.

Wstawiając te dane do wzoru otrzymujemy oszacowanie krytycznej prędkości  $v$ :

$$v = \sqrt{\frac{rsg}{h}} = \sqrt{\frac{(5m)(1.5m)(9.81m/s^2)}{(2m)}} = 36.8 \left(\frac{m}{s}\right)$$

Widzimy, że samochód dostawczy wykonując ostry skręt przy prędkości większej niż 132 km/h (= 36.8 m/s) może się wywrócić.



Rys. 6.15 Metoda pomiaru prędkości obiektu – należy pamiętać o zatrzymaniu stopera, gdy skończy się lot (źródło anonimowe, Internet)





---

# ROZDZIAŁ 7

## MODELOWANIE ZDERZEŃ

---

7.1. Zderzenia obiektów i wykrywanie kolizji. ....	138
7.2. Elementy fizyki zderzeń. ....	139
7.3. Metody wykrywania kolizji. ....	152

---

## 7.1. Zderzenia obiektów i wykrywanie kolizji.

W grach komputerowych istotne jest poprawne modelowanie zderzeń obiektów. Dostępne obecnie procesory (CPU i GPU) pozwalają realistycznie symulować różnego typu zderzenia w czasie rzeczywistym. Termin „*realistyczne symulacje*” odnosi się do zastosowania zasad fizyki, dzięki czemu możemy poprawnie opisać stan obiektów po zderzeniu.

Należy rozróżniać detekcję zderzeń (ang. *collisin detection*) od efektu zderzenia (ang. *collision response*). Programowanie detekcji zderzeń polega na implementacji odpowiednich algorytmów geometrii obliczeniowej w celu wyliczenia miejsca zderzenia i czasu zderzenia. Istnieje wiele inteligentnych i zaawansowanych algorytmów tego typu, należy jednak zwrócić uwagę na opinie praktyków, że zazwyczaj metody brył otaczających dają, jak dotąd najlepsze efekty. Programowanie efektów zderzenia sprowadza się do zastosowania zasad kinetyki i dynamiki w celu określenia stanu obiektów po zderzeniu i znalezienia deskryptorów fizycznych obiektów (prędkość, przyspieszenie, pęd, energia, itp.), tak, aby można było ustalić ich trajektorie.

Według specjalistów, modelowanie zderzeń jest jednym z najtrudniejszych zagadnień. Współczesne gry komputerowe są implementowane w przestrzeni 3D i muszą szybko działać. Optymalizacja algorytmów jest sprawą kluczową. W swojej dyskusji o wykrywaniu kolizji Andre LaMothe przytacza następujący przykład. Jeżeli przyjmiemy, że gra obsługuje 1000 obiektów, aby wykryć kolizje dla tych obiektów należy wykonać około 1 miliona obliczeń. Jeżeli są to obiekty w zaawansowanych grach wojennych (np. walka w kosmosie, poruszają się statki kosmiczne, strzelają działa, wystrzeliwane są rakiety, mamy promienie laserowe, itp.), to liczba obiektów do obsłużenia może wzrosnąć do 10 000. W takim przypadku należy wykonać ponad 100 milionów obliczeń! Nawet najbardziej zaawansowane algorytmy tu nie pomogą. Należy zastosować inne podejście do wykrywania kolizji. Inaczej mówiąc zamiast obliczeń typu „*brute force*”, wykorzystać technikę dzielenia przestrzeni na segmenty. Całą przestrzeń dzielimy na obszary. W takim przypadku liczba obiektów w obszarze rzadko przekroczy, 50 (gdy mamy obiektów więcej, zawsze można dokonać ponownego podziału przestrzeni, aby zmniejszyć obszary). Przy tysiącu obiektów, statystycznie rzecz biorąc, otrzymamy 200 obszarów ( $1000 : 50 = 200$ ), które musimy zanalizować. W każdym obszarze wykonamy około 2500 obliczeń ( $50 \times 50$ ). Biorąc pod uwagę 200 obszarów, ostatecznie należy dokonać 500 000 obliczeń. Jest to znaczna redukcja zapotrzebowania na moce obliczeniowe w porównaniu z pierwotnym zadaniem wykonania 100 milionów obliczeń, gdy nie ma podziału przestrzeni. Stosując segmentację przestrzeni (technicznie tworzy się odpowiednią strukturę danych, np. drzewa ósemkowe lub binarny podział przestrzeni- BSP) redukujemy żadaną liczbę obliczeń do 0.5 %.

Fizyka zderzeń także nie jest prostym zagadnieniem. W podręcznikach fizyki zwykle opisuje się zderzenia punktów materialny, a same zderzenia są sprężyste.

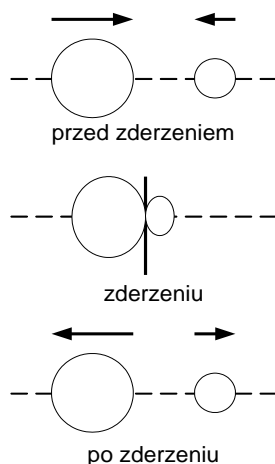
Taki przypadek rzadko występuje w grafice komputerowej i grach komputerowych. Zazwyczaj obiekty mają rozmiary (najczęściej nieregularne) a zderzenia nie są sprężyste. W dodatku podczas zderzeń w wielu przypadkach należy uwzględnić ruch obrotowy, co znakomicie komplikuje opis analityczny. Zderzenia mogą być niecentralne, co jest dodatkową komplikacją.

## 7.2. Elementy fizyki zderzeń.

Formalnie podajemy definicje zderzenia zgodnie z podręcznikiem Hallidaya i innych (Podstawy fizyki, PWN, Warszawa, 2005):

*Zderzenie zachodzi wtedy, gdy dwa lub więcej ciał (partnerów zderzenia) działa na siebie stosunkowo dużymi siłami w stosunkowo krótkim przedziale czasu.*

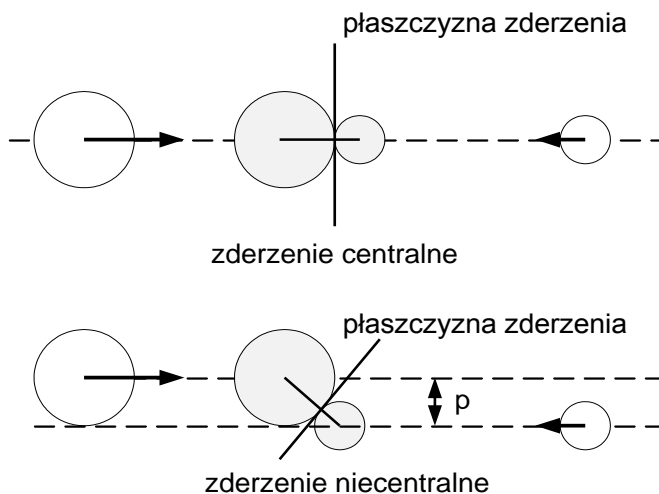
W praktyce, zderzenie rozumiemy, jako zetknięcie się dwóch obiektów. Zjawisko zderzenia się dwóch kul (bardzo schematycznie) pokazane jest na rysunku 7.1.



Rys.7.1 Zderzenia obiektów

Na rysunku pokazano przebieg zderzenia dwóch kul. Kule są jednorodne (środek masy każdej z kul jest w punkcie środkowym kuli), zderzają się centralnie (wektory prędkości obu kul leżą na prostej łączącej ich środki mas), są idealnie gładkie (nie uwzględniamy tarcia), nie występują opory powietrza. Widzimy, że opisujemy w sposób mało realistyczny zjawisko zderzenia. W momencie zetknięcia kule oddziałują w bardzo krótkim okresie czasu bardzo dużymi siłami (te siły są nazywane siłami impulsowymi lub siłami uderzeniowymi). Podczas uderzenia kule ulegają deformacji.

Zderzenia mogą być centralne lub niecentralne (rys.7.2). Ze zderzeniem niecentralnym mamy do czynienia, gdy wektory prędkości kul nie leżą na prostej łączącej ich środki mas.



Rys. 7.2. Zderzenia centralne i niecentralne

Zderzenia dzielimy na elastyczne (sprężyste) i nieelastyczne (niesprężyste). W zderzeniach elastycznych (ang. *elastic collisions*) spełniona jest:

- zasada zachowania pędu
- zasada zachowania energii kinetycznej

W zderzeniach nieelastycznych (ang. *non-elastic collisions*) energia kinetyczna zamieniana jest w ciepło, może także powodować deformacje. W zderzeniach niesprężystych:

- zachowany jest pęd
- energia kinetyczna nie jest zachowana

Przypominamy, że pędem cząstki jest wektor  $p$  :

$$\vec{p} = m\vec{v}$$

gdzie  $m$  jest masą cząstki, a  $v$  jest jej prędkością.

Drugą zasadę dynamiki Newtona możemy zapisać w postaci:

$$\vec{F} = \frac{d\vec{p}}{dt} \quad (1)$$

Jeżeli wykorzystamy równość  $\mathbf{p} = m\mathbf{v}$  to mamy:

$$\vec{F} = \frac{d\vec{p}}{dt} = \frac{d}{dt}(m\vec{v}) = m \frac{d\vec{v}}{dt} = m\vec{a}$$

Jeżeli układ jest izolowany, a żadna cząstka nie opuszcza układu ani nie przybywa nowa cząstka do układu (mówimy, że układ jest zamknięty) a wypadkowa sił działających na układ cząstek jest równa zero to możemy napisać:

$$\vec{F} = \frac{d\vec{p}}{dt} = 0$$

Oznacza to, że:

$$\vec{p} = \text{const}$$

Jest to **zasada zachowania pędu**. Ta zasadę formułujemy w następujący sposób:

*Jeżeli na układ cząstek nie działają siły zewnętrzne lub ich wypadkowa jest równa zero, to całkowity pęd układu cząstek nie ulega zmianie.*

Jeżeli dochodzi do zderzenia dwóch cząstek, to pędy tych cząstek ulegną zmianie. Możemy przekształcić równanie (1) do postaci:

$$d\vec{p} = \vec{F}(t)dt$$

Po scałkowaniu tego równania po czasie otrzymamy:

$$\int_{p_1}^{p_2} d\vec{p} = \int_{t_1}^{t_2} \vec{F}(t)dt$$

Z lewej strony równania mamy wyrażenie  $(p_2 - p_1)$ , czyli zmianę pędu. Prawa strona równania jest tzw. popędem siły:

$$\vec{J} = \int_{t_1}^{t_2} \vec{F}(t)dt$$

Zmiana pędu cząstki podczas zderzenia jest równa popędowi siły:

$$\vec{p}_2 - \vec{p}_1 = \Delta\vec{p} = \vec{J}$$

Energia kinetyczna  $E_k$  cząstki o masie  $m$  jest związane z jej prędkością  $v$ :

$$E_k = \frac{1}{2}mv^2$$

Gdy prędkość jest równa zero to energia kinetyczna też jest równa zero. Jednostką energii kinetycznej jest dżul (J). Z definicji energii kinetycznej  $E_k$  wynika wymiar dżula:

$$1 \text{ dżul} = 1 \text{ J} = 1 \text{ kg} \frac{\text{m}^2}{\text{s}^2}$$

Oprócz energii kinetycznej wyróżniamy energię potencjalną  $E_p$ . Energia potencjalna jest to energia związana z konfiguracją układu ciał, działających na siebie siłami.

Szkolnym przykładem energii potencjalnej jest grawitacyjna energia potencjalna. Gdy siła  $F$  wykonuje nad ciałem pracę  $W$  (np. kamień jest podnoszony), to zmienia się energia potencjalna układu:

$$\Delta E_p = -W$$

Bardziej formalnie, gdy siła jest zachowawcza i wykonuje pracę nad ciałem przemieszczając go punktu początkowego  $x_p$  do punktu końcowego  $x_k$ , zmiana energii potencjalnej wyraża się wzorem:

$$\Delta E_p = - \int_{x_p}^{x_k} F(x) dx$$

Grawitacyjna energia potencjalna  $E_p$  wyraża się wzorem:

$$E_p = mgh$$

gdzie  $g$  jest to stała grawitacyjna,  $m$  jest masą ciała a  $h$  jest wysokością (liczoną od punktu odniesienia, może to być poziom morza).

Energia mechaniczna  $E_m$  układu jest to suma energii kinetycznej i potencjalnej:

$$E_m = E_k + E_p$$

Gdy układ jest izolowany oraz działają jedynie siły zachowawcze, to spełniona jest **zasada zachowania energii mechanicznej**. Energia kinetyczna i potencjalna mogą się zmieniać, ale ich suma musi postawać stała:

$$\Delta E_m = \Delta E_k + \Delta E_p = 0$$

Dzięki zasadzie zachowania energii możemy w sposób stosunkowo prosty wyliczać różne wielkości charakteryzujące ciała biorące udział w zderzeniach. Oprócz energii potencjalnej, wprowadzamy do opisu zjawisk fizycznych inne rodzaje energii, na przykład energię termiczną. Możemy wprowadzić pojęcie całkowitej energii układu  $E$ . Jest to suma energii kinetycznej, potencjalnej, termicznej  $E_t$  oraz innych rodzajów tzw. energii wewnętrznej  $E_w$ . Obowiązuje zasada zachowania energii całkowitej:

$$\Delta E = \Delta E_m + \Delta E_t + \Delta E_w$$

Zmiana całkowitej energii układu  $E$  jest równa energii dostarczonej do układu lub od niego odebranej. Mówiąc wprost – energia nie może powstawać z niczego ani nie może znikać!

W układzie izolowanym całkowita energia  $E$  nie może się zmieniać:

$$\Delta E_m + \Delta E_t + \Delta E_w = 0$$

Przypomnimy jeszcze pojęcie pracy i mocy.

Formalnie praca  $W$  jest to energia przekazana ciału lub od niego odebrana na drodze działania na ciało siłą. Praca  $W$  jest równa zmianie energii.

Praca wykonana przez stałą siłę wyraża się wzorem:

$$W = \vec{F} \cdot \vec{d}$$

gdzie  $F$  jest siłą a  $d$  jest wektorem przesunięcia ciała.

Jednostką pracy jest dżul (tak samo jak energii kinetycznej).

Szybkość, z jaką siła wykonuje pracę nazywana jest mocą  $P$ :

$$P = \frac{dW}{dt}$$

Jednostka mocy jest wat (W), wymiar tej jednostki to dżul na sekundę:

$$1 W = 1 \frac{J}{s} = 1 \frac{kg \cdot m^2}{s^3}$$

Używana jest także stara jednostka mocy zwana koniem mechanicznym (KM):

$$1 KM = 746 W$$

Podczas zderzenia działają bardzo duże siły w bardzo krótkim czasie. Popęd siły (ang. *impulse*) może osiągnąć duże wartości. Rozważmy krążek hokejowy o masie 100 g leżący na lodzie (np. opisujemy grę w hokeja). Pomiar pokazał, że uderzony kijem hokejowym krążek był w kontakcie z kijem przez około 1 milisekundę i osiągnął prędkość  $v_2 = 30$  m/s. Jaki był popęd i jaka była wartość siły działającej na krążek?

Popęd siły  $J$  równy jest zmianie pędu ( $v_1 = 0$ ):

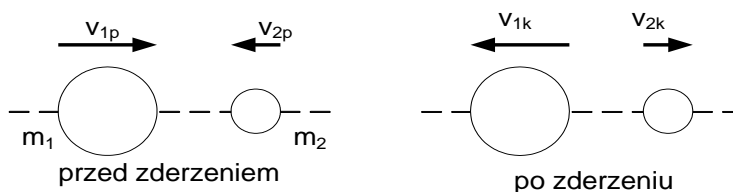
$$J = p_2 - p_1 = mv_2 - mv_1 = 0.1(\text{kg}) \cdot 30 \left(\frac{\text{m}}{\text{s}}\right) - 0 \left(\frac{\text{m}}{\text{s}}\right) = 3.0 \left(\frac{\text{kg} \cdot \text{m}}{\text{s}}\right)$$

W celu obliczenia siły działającej na krążek należy podzielić popęd  $J$  siły przez czas działania siły  $\Delta t$ :

$$F = \frac{J}{\Delta t} = \frac{3.0 \left(\frac{\text{kg} \cdot \text{m}}{\text{s}}\right)}{10^{-3}(\text{s})} = 3000.0 \text{ N}$$

Jak widzimy jest to bardzo duża wartość siły, nic dziwnego, że często kije hokejowe pękają podczas uderzenia w krążek.

Praktycznie, w życiu codziennym, nigdy nie mamy do czynienia ze zderzeniami sprężystymi. Zawsze część energii kinetycznej będzie tracona przy zderzeniach. Nawet zderzenia kul bilardowych, które w zasadzie traktujemy, jako przykład zderzeń sprężystych, nie są takimi. Podczas zderzenia słychać trzask uderzanych kul, co oznacza, że powstała fala dźwiękowa, a to oznacza, że część energii kinetycznej została zamieniona w energię fali dźwiękowej. W symulacjach chętnie korzystamy z przybliżenia zderzeń sprężystych, ponieważ parametry ruchu dają się stosunkowo prosto wyliczyć, a błędy oszacowań tych parametrów nie są zbyt duże. Na rys. 7.3 schematycznie przedstawiono zderzenie centralne, sprężyste dwóch ciał.



Rys.7.3. Zderzenie centralne sprężyste

Ciało o masie  $m_1$  porusza się z prędkością  $v_{1p}$  w kierunku ciała o masie  $m_2$ , które porusza się z prędkością  $v_{2p}$ . Po zderzeniu ciała poruszają się z prędkościami  $v_{1k}$  i  $v_{2k}$ .



Zasada zachowania pędu ma postać:

$$m_1 v_{1p} + m_2 v_{2p} = m_1 v_{1k} + m_2 v_{2k}$$

Zasada zachowania energii kinetycznej ma postać:

$$\frac{1}{2} m_1 v_{1p}^2 + \frac{1}{2} m_2 v_{2p}^2 = \frac{1}{2} m_1 v_{1k}^2 + \frac{1}{2} m_2 v_{2k}^2$$

Z tego układu równań możemy wyliczyć prędkości końcowe dwóch ciał po zderzeniu:

$$v_{1k} = \frac{m_1 - m_2}{m_1 + m_2} v_{1p} + \frac{2m_2}{m_1 + m_2} v_{2p}$$

$$v_{2k} = \frac{m_2 - m_1}{m_1 + m_2} v_{2p} + \frac{2m_1}{m_1 + m_2} v_{1p}$$

Rozważymy kilka przypadków szczególnych.

Jeżeli ciało o masie  $m_2$  jest nieruchome ( $v_{2p} = 0$ ) to po zderzeniu prędkości kul są dane wzorami:

$$v_{1k} = \frac{m_1 - m_2}{m_1 + m_2} v_{1p}$$

$$v_{2k} = \frac{2m_1}{m_1 + m_2} v_{1p}$$

Jeżeli masy ciał są jednakowe (tak jest na przykład, gdy modelujemy zderzenia kul bilardowych) to wtedy prędkości po zderzeniu są następujące:

$$v_{1k} = 0$$

$$v_{2k} = v_{1p}$$

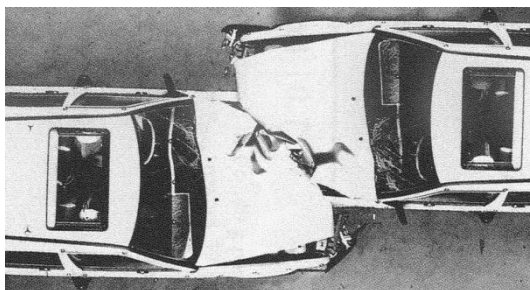
Jest to idealne zderzenie kul bilardowych – po zderzeniu poruszająca się kula zatrzymuje się, a nieruchoma kula uzyskuje prędkość kuli ruchomej. Mówimy, że dwa ciała „wymieniły się prędkościami”.

W zderzeniu centralnym niesprężystym spełniona jest jedynie zasada zachowania pędu:

$$m_1 v_{1p} + m_2 v_{2p} = m_{1k} v_{1k} + m_2 v_{2k}$$

Gdy znane są masy ciał i ich prędkości przed zderzeniem oraz np. prędkość ciała o masie  $m_1$  po zderzeniu niesprężystym, to wtedy możemy wyliczyć prędkość ciała o masie  $m_2$ .

Dość często zdarzają się zderzenia całkowicie niesprężyste. Praktycznie zawsze mamy do czynienia ze zderzeniami niesprężystymi. Przykład takiego zderzenia pokazany jest na rysunku 7.4. Jest to fotografia wypadku samochodowego, w którym dwa samochody zderzyły się prawie czołowo. Widach wyraźnie odkształcenia karoserii – cała energia kinetyczna wykorzystana została do deformacji blachy.

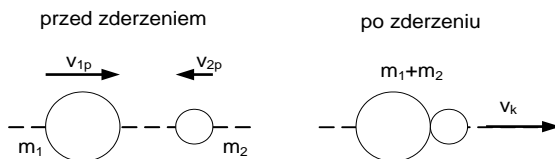


Rys. 7.4. Zderzenie dwóch samochodów, zderzenie niesprężyste

Gdy ciało o masie  $m_2$  jest nieruchome ( $v_{2p} = 0$ ) to prędkość końcowa w zderzeniu całkowicie sprężystym wyraża się wzorem:

$$v_k = \frac{m_1}{m_1 + m_2} v_{1p}$$

Tego typu zderzenie pokazane jest na rysunku 7.5. Ciało o masie  $m_2$  jest nieruchome, po zderzeniu oba ciała łączą się i poruszają się z prędkością  $v_k$ .



Rys. 7.5. Zderzenie centralne, niesprężyste

W zderzeniach niecentralnych, wektory prędkości dwóch ciał nie leżą na linii łączącej ich środki mas, zderzenia musimy opisać w dwóch wymiarach. Rozważmy przypadek, gdy zderzenie jest niecentralne, ciało o masie  $m_1$  porusza się z prędkością  $v_{1p}$  a ciało o masie  $m_2$  jest nieruchome (rys. 7.6). Tor ciała o masie  $m_1$  po zderzeniu tworzy kąt  $\phi_1$  z kierunkiem ruchu ciała przed zderzeniem.

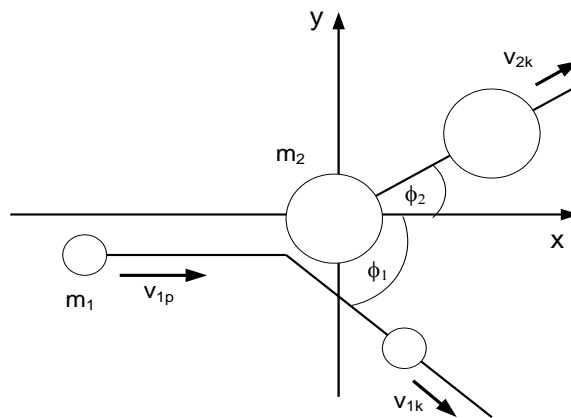
Ciało o masie  $m_2$ , które jest początkowo nieruchome, po zderzeniu porusza się po torze, którego kierunek tworzy kąt  $\phi^2$  z osią  $x$ . W tego typu zderzeniach zawsze jest spełniona zasada zachowania pędu. Możemy ułożyć równania dla składowej  $x$ :

$$m_1 v_{1p} = m_1 v_{1k} \cos \phi_1 + m_2 v_{2k} \cos \phi_2$$

oraz składowej  $y$

$$0 = m_1 v_{1k} \sin \phi_1 - m_2 v_{2k} \sin \phi_2$$

Te dwa równania związane z zasadą zachowania pędu są zawsze spełnione – zarówno dla zderzeń sprężystych jak i dla zderzeń niesprężystych.

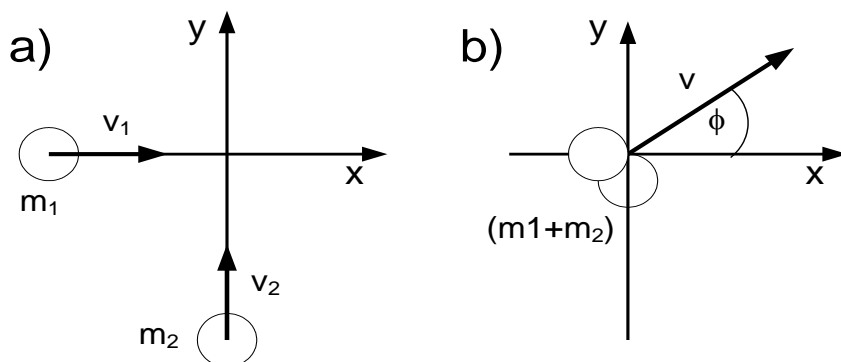


Rys. 7.6. Zderzenie niecentralne

Jeżeli zderzenie jest sprężyste to spełniona jest zasada zachowania energii kinetycznej:

$$\frac{1}{2} m_1 v_{1p}^2 = \frac{1}{2} m_1 v_{1k}^2 + \frac{1}{2} m_2 v_{2k}^2$$

Pokazane równania opisujące zderzenia niecentralne mogą wydawać się bardzo skomplikowane. Pokażemy jak powyższe wzory możemy wykorzystać do opisu zderzenia dwóch łyżwiarzy. Rozważmy dwóch łyżwiarzy poruszających się na lodowisku. Tory łyżwiarzy tworzą kąt prosty (rys. 7.7a). Po chwili łyżwiarze zderzają się i złączeni kontynuują ruch w kierunku wypadkowym (rys. 7.7).



Rys.7.7. Schematycznie przedstawione zderzenie dwóch łyżwiarzy, zderzenie niesprężyste, a) – sytuacja przed zderzeniem, b) – sytuacja po zderzeniu

Wiemy, że masy łyżwiarzy są odpowiednio  $m_1=60$  kg,  $m_2=45$  kg, prędkości są  $v_1=6$  km/h i  $v_2=10$  km/h. Po zderzeniu łyżwiarze kontynuują ruch z prędkością  $v$ . Możemy wyliczyć tą prędkość oraz kąt, pod jakim będą kontynuować ruch złączeni.

Jeżeli potraktujemy układ łyżwiarzy, jako układ izolowany to możemy zastosować zasadę zachowania pędu:

$$m_1\vec{v}_1 + m_2\vec{v}_2 = (m_1 + m_2)\vec{v}$$

Prędkość końcowa dana jest wzorem:

$$\vec{v} = \frac{m_1\vec{v}_1 + m_2\vec{v}_2}{(m_1 + m_2)}$$

Są to równania wektorowe. Możemy wyliczyć prędkość i kąt ruchu po zderzeniu, jeżeli wypiszemy równania pędu wzdłuż każdej osi współrzędnych.

Równanie na składową  $x$  pędu ma postać:

$$m_1v_1 = (m_1 + m_2)v \cos \varphi$$

Analogicznie wyrażenie na składową  $y$  pędu ma postać:

$$m_2v_2 = (m_1 + m_2)v \sin \varphi$$

Dziela te równania przez siebie i wstawiając dane liczbowe mamy:

$$\tan \varphi = \frac{m_2 v_2}{m_1 v_1} = \frac{45 \text{ kg} \cdot 10 \frac{\text{km}}{\text{h}}}{60 \text{ kg} \cdot 6 \frac{\text{km}}{\text{h}}} = 1.25$$

Kąt  $\varphi$  wynosi około  $51^\circ$ . Wykorzystując składową  $y$  pędu możemy obliczyć wartość prędkości:

$$v = \frac{m_2 v_2}{(m_1 + m_2) \sin \varphi} = \frac{45 \text{ kg} \cdot 10 \frac{\text{km}}{\text{h}}}{(60 \text{ kg} + 45 \text{ kg}) \sin 51^\circ} = 5.51 \frac{\text{km}}{\text{h}}$$

Możemy także oszacować, jaka część energii kinetycznej została zamieniona na energię, którą musiały przyjąć na siebie ciała łyżwiarzy podczas zderzenia. Przed zderzeniem, energia kinetyczna łyżwiarzy wynosiła:

$$E_p = \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2$$

Po podstawieniu wartości ( prędkości wyrażamy w jednostkach m/s) mamy:

$$E_p = 0.5 \cdot (60 \text{ kg} \cdot (1.7 \text{ m/s})^2 + 45 \text{ kg} \cdot (2.8 \text{ m/s})^2) = 263.1 \text{ J}$$

Podobnie możemy obliczyć energię końcową:

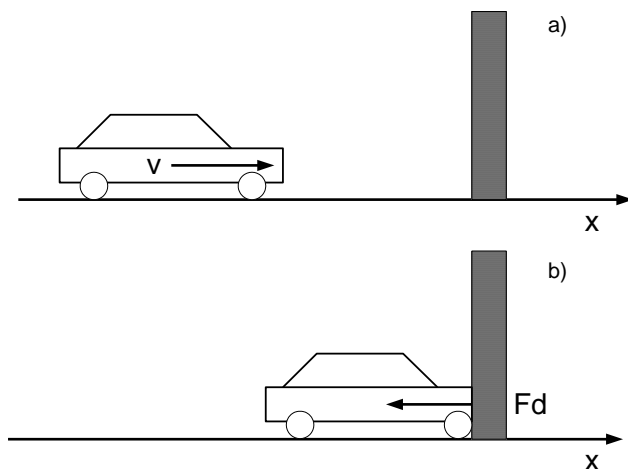
$$E_k = \frac{1}{2} (m_1 + m_2) v^2$$

Po podstawieniu wartości liczbowych mamy:

$$E_k = 0.5 \cdot (60 \text{ kg} + 45 \text{ kg}) \left(1.53 \frac{\text{m}}{\text{s}}\right)^2 = 122.8 \text{ J}$$

Widzimy, że energia kinetyczna po zderzeniu znacznie zmalała, około 50 % początkowej energii kinetycznej łyżwiarzy ulega rozproszeniu w momencie zderzenia.

Wypadki z udziałem samochodów, gdy dochodzi do zderzenia, są bardzo niebezpieczne ze względu na dużą prędkość i dużą masę pojazdu, energia tracona w takim przypadku jest bardzo duża. Wyliczymy, jaka siła działa na samochód podczas jego zderzenia z nieruchomą przeszkodą.



Rys.7.8. Zderzenie niesprężyste samochodu z przeszkodą

Samochód o masie  $m$  porusza się z prędkością  $v$ , w pewnym momencie wpada na betonowy słup. Jaki był efekt zderzenia, jeżeli odkształcenie było równe  $d$  metrów?

Praca potrzebna do zatrzymania samochodu jest dana wzorem:

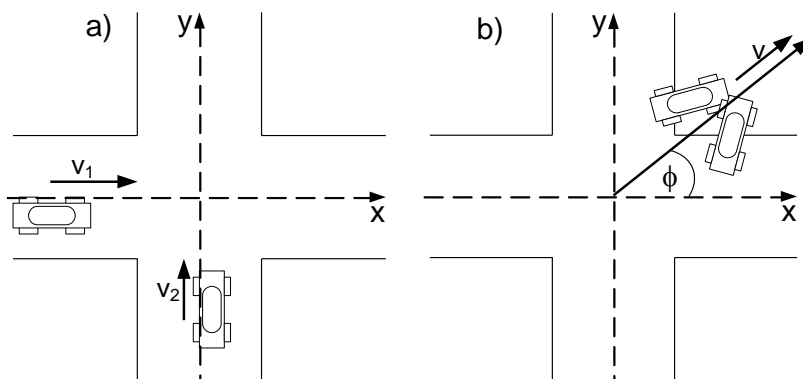
$$F_{av}d = -\frac{mv^2}{2}$$

gdzie  $F_{av}$  jest to średnia siła działająca podczas zderzenia,  $d$  jest długością odkształcenia. Możemy wyznaczyć z tego równania siłę  $F_{av}$ :

$$F_{av} = -\frac{mv^2}{2d}$$

Wstawiając odpowiednie wielkości możemy przekonać się, jak duże siły działają podczas takiego zderzenia.

Znajomość zasady fizyki pozwala na rozstrzygnięcie czy uczestnicy wypadku samochodowego mówią prawdę. Rozważmy następujący przykład. Na skrzyżowaniu ulic dochodzi do kolizji dwóch samochodów. Kierowca pierwszego samochodu jedzie w kierunku  $x$  z prędkością  $v_1 = 47$  km/h, drugi kierowca jedzie z prędkością  $v_2$  w kierunku  $y$  (rys.7.9.a).



Rys.7.9. Zderzenie samochodów na skrzyżowaniu

Następuje kolizja, samochody zakleszczają się i razem przesuwane są w kierunku tworzącym z osią  $x$  kąt  $\phi = 55^{\circ}$  (rys. 7.9b). Dozwolona prędkość na skrzyżowaniu to 50 km/h. Kierowca samochodu jadącego w kierunku  $y$  twierdzi, że nie przekroczył dopuszczalnej prędkości. Czy mówił prawdę? Ułożymy równania dla składowych  $x$  i  $y$  pędów:

$$mv_1 = 2mv_k \cos \varphi \quad (\text{kierunek } x)$$

$$mv_2 = 2mv_k \sin \varphi \quad (\text{kierunek } y)$$

Dzieląc te równania stronami, wyznaczając  $v_2$  oraz wstawiając dane liczbowe otrzymujemy:

$$v_2 = v_1 \tan \varphi = 47 \frac{\text{km}}{\text{h}} \cdot \tan 55.0 = 67.1 \text{ km/h}$$

Na podstawie modelowania wypadku samochodowego, uważamy, że drugi kierowca, jadący w kierunku  $y$  przekroczył dozwoloną prędkość.

Dwie zasady – zasada zachowania pędu i zasada zachowania energii całkowitej są spełnione w zderzeniach. Energia kinetyczna jest zachowana tylko w zderzeniach sprężystych. Prawdę mówiąc, w otaczającym nas świecie zderzenia sprężyste prawie nie zachodzą. W rzeczywistości zawsze mamy do czynienia ze stratami energii. Możemy określić współczynnik strat energii wprowadzając tzw. współczynnik restytucji  $e$  (ang. *coefficient of restitution*):

$$e = -\frac{v_{1k} - v_{2k}}{v_{1p} - v_{2p}}$$

Współczynnik restytucji jest stosunkiem względnej prędkości ciał po zderzeniu do względnej prędkości przed zderzeniem. Współczynnik restytucji zależy od materiału, budowy i geometrii zderzających się ciał. Współczynnik restytucji  $e$  wyznaczany jest doświadczalnie. Gdy zderzenia są idealnie sprężyste, współczynnik restytucji  $e$  równy jest jeden, gdy mamy do czynienia ze zderzeniami idealnie niesprężystymi to współczynnik restytucji równy jest zeru. W przypadkach pośrednich  $0 < e < 1$ . Jeżeli ciało spada swobodnie i uderza w podłogę, to współczynnik restytucji  $e$  wyraża się prostym wzorem:

$$e = \sqrt{\frac{h}{H}}$$

gdzie  $H$  oznacza wysokość, z jakiej spada ciało, a  $h$  jest wysokością, na jaką odbije się ciało. W tabeli 7.1. pokazano współczynniki restytucji.

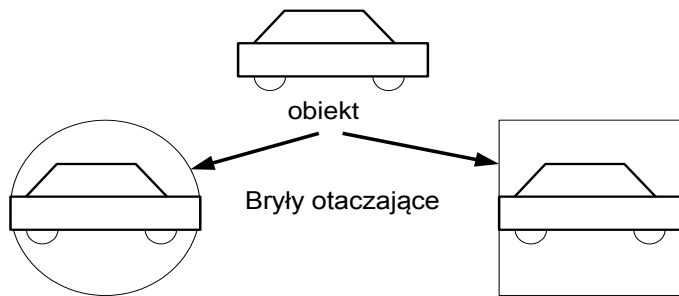
Ciało	Współczynnik restytucji $e$
Piłka golfowa	0.86
Piłka tenisowa	0.71
Kula bilardowa	0.80
Piłka do siatkówki	0.75
Kula drewniana	0.60
Kulka stalowa	0.60

Znajomość współczynnika restytucji pomaga wyznaczać parametry ruchu ciał po zderzeniu.

### 7.3. Metody wykrywania kolizji.

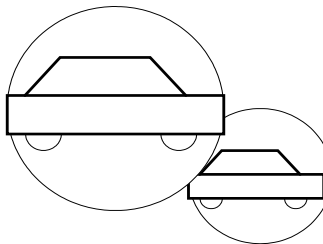
Istotnym zagadnieniem jest stwierdzenie faktu czy doszło do kolizji. Wykrywanie kolizji w grach komputerowych nie jest trywialnym zagadnieniem. Dysponujemy wieloma technikami wykrywania kolizji. Jednym z najprostszych sposobów wykrywania kolizji jest zastosowanie techniki brył otaczających. W tym celu w przestrzeni 3D najczęściej wykorzystujemy sześcian lub sferę, a w przestrzeni 2D wykorzystujemy kwadrat lub okrąg (rys. 7.10).





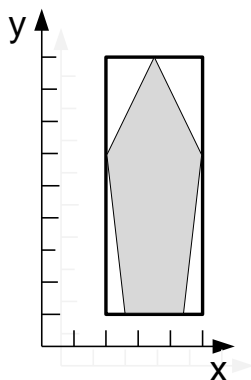
Rys.7.10 Bryły otaczające – okrąg i kwadrat

Promień okręgu dobieramy w ten sposób, aby wszystkie elementy obiektu były w nim zamknięte. Technicznie oznacza to, że przeglądamy wszystkie wierzchołki obiektu i sprawdzamy czy są wewnątrz okręgu. Podobnie postępujemy znajdując kwadrat otaczający obiekt. Gdy dla obiektów zostaną znalezione okręgi otaczające (położenie środka okręgu i jego promień), wykrycie kolizji jest już zadaniem prostym. Należy ustalić, czy spełniony jest warunek, że odległość pomiędzy środkami dwóch okręgów jest mniejsza niż suma ich promieni. Na rysunku 7.11 pokazana jest sytuacja, gdy znaleziona jest kolizja.



Rys. 7.11 Wykrywanie zderzenia przy pomocy okręgów otaczających

W wielu przypadkach zastosowanie prostokątów otaczających lub prostopadłościanów może dać dokładniejsze wyniki przy wykrywaniu kolizji, gdyż bryła otaczająca lepiej przybliży kształty obiektu. Taka sytuacja pokazana jest na rys. 7.12. Zastosowanie okręgu ograniczającego obiekt byłoby zbyt dużym przybliżeniem. W celu utworzenia prostokąta ograniczającego należy przejrzeć listę wierzchołków obiektu i wyszukać wierzchołków o największej i najmniejszej współrzędnej  $x$  i  $y$ . Funkcja wyznaczająca prostokąt ograniczający może mieć postać pokazana na listingu 7.1. Prostokąt otaczający zdefiniowany jest dwoma wierzchołkami  $LD(x,y)$  oraz  $PG(x,y)$ , gdzie  $LD$  jest to dolny prawy wierzchołek prostokąta a  $PG$  jest to prawy górny wierzchołek prostokąta.



Rys.7.11. Prostokąt ograniczający obiektu

## Listing 7.1. Prostokąt otaczający, testowanie kolizji

---

```

#include <iostream>
#include <conio>
using namespace std;
#define NP 5
struct point {int x, y; };

int main()
{struct point ob[NP]=
  {{5,2}, {4,12}, {7,18}, {10,12}, {9,2} };
  struct point ob1 = {12,5}; //testowany punkt
  int minx=ob[0].x;
  int miny=ob[0].y;
  int maxx=ob[0].x;
  int maxy=ob[0].y;

  for (int i=1; i<NP; i++)
  {if (ob[i].x < minx) minx = ob[i].x;
    if (ob[i].y < miny) miny = ob[i].y;
    if (ob[i].x > maxx) maxx = ob[i].x;
    if (ob[i].y > maxy) maxy = ob[i].y;
  }
  // wykrywanie kolizji:
  if((ob1.x >=minx && ob1.y >= miny) &&
    (ob1.x <= maxx && ob1.y <= maxy))
    cout <<" jest kolizja";
  else
    cout <<" nie ma kolizji";
  getch();
  return 0;
}

```

---

W programie przy pomocy pętli *for* przeglądamy wierzchołki obiektu *ob* i wyznaczamy współrzędne wierzchołków prostokąta otaczającego (*(minx, miny)* oraz *(maxx, maxy)*). W naszym przykładzie otrzymamy wartości LD(4,2) i PG(10,18). Sprawdzamy czy punkt *ob1* o współrzędnych (12,5) znajduje się wewnątrz prostokąta otaczającego obiekt *ob*. W tym przypadku nie jest to prawda i otrzymujemy komunikat „*nie ma kolizji*”. Gdyby punkt *ob1* miał współrzędne (5,5) otrzymalibyśmy komunikat „*jest kolizja*”.

Wykrywanie kolizji, jak wspominaliśmy, jest jednym z podstawowych zadań, które należy rozwiązać w celu prawidłowej prezentacji oddziaływań pomiędzy obiektami istniejącymi w grach komputerowych. Najczęstsze zadanie tego typu to znajdowanie punktu przecięcia dwóch odcinków, punktów przecięcia dwóch okręgów lub kul, itp. Opracowywaniem wydajnych algorytmów rozwiązujących te zadania zajmuje się dział matematyki nazywany *geometrią obliczeniową*. Znajomość algorytmów geometrycznych jest niezbędna do tworzenia wydajnych aplikacji w grach komputerowych.

W monografii K. Loudona znajduje się rozdział poświęcony algorytmom geometrycznym. Autor monografii omawia wydajny algorytm sprawdzania czy dwa odcinki się przecinają. Na listingu 7.2 jest pokazany jest program sprawdzający czy odcinki się przecinają, metodę podał Loudon.

Listing 7.2. Test przecinania się odcinków, algorytm Loudona

```
#include <iostream>
#include <conio.h>
using namespace std;

#define MIN(x, y) (((x) < (y)) ? (x) : (y))
#define MAX(x, y) (((x) > (y)) ? (x) : (y))

typedef struct Punkt
{ double x,y;
} P;

int linia(P p1, P p2, P p3, P p4);
int main()
{P p1,p2,p3,p4;
 P ob1[2]= {{1,1}, {7,6}};
 P ob2[2]= {{2,8}, {6,1}};
 p1 = ob1[0];
 p2 = ob1[1];
 p3 = ob2[0];
 p4 = ob2[1];
 cout << "wynik = " << linia(p1,p2,p3,p4);
```

```

    getche();
    return 0;
}

int linia(P p1, P p2, P p3, P p4)
{ double z1, z2, z3, z4;
  int t1, t2, t3, t4;

  if (!(MAX(p1.x, p2.x) >= MIN(p3.x, p4.x) && MAX(p3.x, p4.x)
    >= MIN(p1.x, p2.x) && MAX(p1.y, p2.y) >= MIN(p3.y, p4.y)
    && MAX(p3.y, p4.y) >= MIN(p1.y, p2.y)))
    return 0;

  if ((z1 = ((p3.x - p1.x)*(p2.y - p1.y)) - ((p3.y - p1.y)*(p2.x - p1.x))) < 0)
    t1 = -1;
  else if (z1 > 0)
    t1 = 1;
  else
    t1 = 0;
  if ((z2 = ((p4.x - p1.x)*(p2.y - p1.y)) - ((p4.y - p1.y)*(p2.x - p1.x))) < 0)
    t2 = -1;
  else if (z2 > 0)
    t2 = 1;
  else
    t2 = 0;
  if ((z3 = ((p1.x - p3.x)*(p4.y - p3.y)) - ((p1.y - p3.y)*(p4.x - p3.x))) < 0)
    t3 = -1;
  else if (z3 > 0)
    t3 = 1;
  else
    t3 = 0;
  if ((z4 = ((p2.x - p3.x)*(p4.y - p3.y)) - ((p2.y - p3.y)*(p4.x - p3.x))) < 0)
    t4 = -1;
  else if (z4 > 0)
    t4 = 1;
  else
    t4 = 0;
  if((t1*t2 <= 0) && (t3*t4 <= 0))
    return 1;
  return 0;
}

```

---

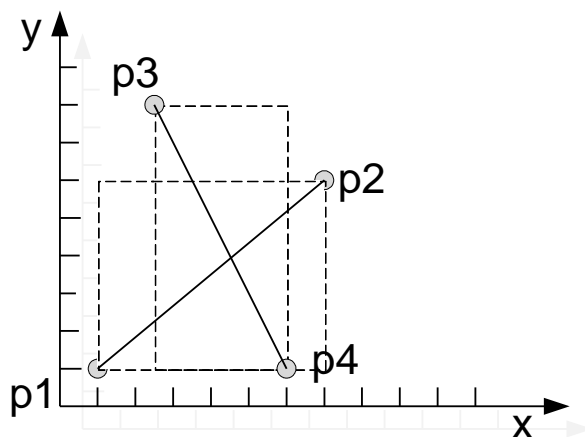
Algorytm jest zoptymalizowany pod kątem szybkości wykonania i dokładności (są oczywiście inne, dużo prostsze algorytmy, nie są one jednak tak wydajne).

Odcinki są definiowane przy pomocy punktów początkowych i końcowych. Odcinek pierwszy definiowany jest parą punktów  $p1(x1,y1)$  oraz  $p2(x2,y2)$  odcinek drugi definiowany jest parą punktów  $p3(x2,y3)$  i  $p4(x4,y4)$ . Program został opracowany dla przypadku 3D, my rozpatrujemy zadanie ustalenia przecięcia dwóch odcinków na płaszczyźnie. W teście rozważamy dwa prostokąty otaczające (rys.7.12). Algorytm jest dwuczęściowy – najpierw wykonywany jest szybki test odrzucenia, a potem wykonywany jest właściwy test.

Krótki test odrzucenia ma postać:

```
if (!(MAX(p1.x, p2.x) >= MIN(p3.x, p4.x) && MAX(p3.x, p4.x)
    >= MIN(p1.x, p2.x) && MAX(p1.y, p2.y) >= MIN(p3.y, p4.y)
    && MAX(p3.y, p4.y) >= MIN(p1.y, p2.y)))
    return 0;
```

Według K. Loudona złożoność funkcji *linia()* wynosi  $O(1)$ , gdyż wszystkie kroki wykonywane są w stałym czasie.



Rys.7.12. Algorytm Loudona do sprawdzenia czy odcinki się przecinają

Funkcja *linia()* zwraca wartość 1, gdy odcinki się przecinają, w przeciwnym przypadku zwraca wartość 0. W pokazanym przykładzie odcinki przecinają się.



## BIBLIOGRAFIA

- [1] B. Stroustrup, *Programowanie, teoria i praktyka z wykorzystaniem C++*, Wydawnictwo Helion, Gliwice, 2010.
- [2] B.W. Kernighan, D.M. Ritchie, *Język ANSI C*, Wydawnictwo Naukowo-Techniczne, Warszawa, 2003.
- [3] G. M. Fichtenholz , *Rachunek różniczkowy i całkowy*, tom 1,2 i 3, wydanie XI, Państwowe Wydawnictwo Naukowe, Warszawa, 1997
- [4] B. Baron, *Metody numeryczne w C++ Builder*, Wydawnictwo Helion, Gliwice, 2004
- [5] R. W. Hornbeck, *Numerical Methods*, Quantum Publishers, Inc, New York, 1975
- [6] R. Graham, D. Knuth, O. Patashnik, *Matematyka konkretna*, PWN, Warszawa, 1996
- [7] R. Parent, *Animacja komputerowa, algorytmy i techniki*, PWN, Warszawa, 2012
- [8] J. Matulewski, T. Dziubak, M. Sylwestrzak, R. Płoszajczak, *Grafika, fizyka, metody numeryczne*, PWN, 2010
- [9] D.M.Bourg, *Fizyka dla programistów gier*, Helion, Gliwice, 2003
- [10] R. Resnick, D. Halliday, *Fizyka*, tom 1, PWN, Warszawa, 1994
- [11] D. Halliday, R. Resnick, J. Walker, *Podstawy fizyki*, tom 1, PWN, 2005
- [12] M. Matyka, *Symulacje komputerowe w fizyce*, Helion, Gliwice, 2002
- [13] A. Stasiewicz, *C++ Builder, 20 efektywnych programów*, Helion, Gliwice, 2002,
- [14] M. Zawacki, *Fizyka, ćwiczenia praktyczne*, Helion, Gliwice, 2002,

