

Jacek RACZKOWSKI, Michał SOBIECKI

POLITECHNIKA WARSZAWSKA, INSTYTUT INFORMATYKI,  
ul. Nowowiejska 15/19, 00-665 Warszawa

## Modelowanie zjawiska wzbitego puchu śnieżnego dla potrzeb animacji komputerowej

Dr inż. Jacek RACZKOWSKI

Absolwent Politechniki Warszawskiej, adiunkt w Instytucie Informatyki Politechniki Warszawskiej. Specjalizuje się w metodach i algorytmach grafiki komputerowej, a w szczególności w modelowaniu i wizualizacji zjawisk naturalnych.



e-mail: j.raczkowski@ii.pw.edu.pl

Mgr inż. Michał SOBIECKI

Absolwent Politechniki Warszawskiej. W swojej pracy dyplomowej opracowywał algorytmy wizualizacji wybranych zjawisk naturalnych. Zawodowo zajmuje się programowaniem grafiki dla gier komputerowych.



e-mail: michal.sobiecki@cdprojektred.com

### Streszczenie

W pracy zaproponowano metodę tworzenia symulatora przepływu płynu i wykorzystania utworzonego modelu do realistycznej wizualizacji wzbitego puchu śnieżnego. Symulator bazuje na uproszczonym i stabilnym rozwiązaniu równań Naviera-Stokesa w przestrzeni wokselowej. Przedstawiony został wpływ poszczególnych składników równań (adwekcja, dyfuzja, wpływ sił zewnętrznych i ciśnienia) na uzyskiwany rozkład gęstości płynu. W pracy opisano przykładowe animacje i przedstawiono analizę wydajności rozwiązania.

**Słowa kluczowe:** symulacja płynów, modelowanie zjawisk naturalnych, animacja komputerowa, śnieg.

### Modeling the phenomenon of entrained powder snow for the computer animation

#### Abstract

Computer animation combines art and engineering in order to create artificial but realistic images. It is widely used in movie industry, advertisement, and in all areas where visual imaging is applied. Modeling natural phenomena including fluid flows is an important part of this task.

In our paper we propose a method of simulation and rendering of fluid to get realistic images of selected natural phenomena. We concentrate on an animation of powder snow and pretend to produce video sequences of entrained snow by a skier or a snowboarder, snow making devices, and snow blown off from a roof. The model of snow movement is based on Navier-Stokes equations solved in an unconditionally stable way in the voxel space. All factors effecting fluid movement, including advection, external forces, diffusion and pressure impact, are considered. We propose the method for calculations of these factors assuming a number of simplifications in our solver. The volume ray-tracer from the Blender 3D open-source visualization environment is applied to render subsequent frames. As a result we present examples of created animations showing the behavior of entrained snow. This is to prove that our solver can produce accepted visual reality of the considered phenomenon. We analyze the efficiency of the proposed modeling method concluding that some factors may significantly reduce the computing power and memory requirements.

**Keywords:** fluid simulation, modeling of natural phenomena, computer animation, snow.

### 1. Cel pracy

Animacja komputerowa jako dziedzina łącząca sztukę z inżynierią jest szeroko stosowana w filmach, reklamach i grach komputerowych. Twórcy zajmujący się animacją często dążą do osiągnięcia jak największego realizmu, szczególnie w obszarze zjawisk naturalnych. Całą domenę takich zjawisk stanowi symulacja płynów, która jest podstawą do ogromnej liczby efektów specjalnych w superprodukcjach kinowych czy najlepszych grach.

Już w latach 80 pojawiały się publikacje na temat modelowania tego typu zjawisk, dziedzina ta nabrała jednak prawdziwego rozpędu pod koniec lat 90. Momentem przełomowym wydają się być

publikacje artykułów Rolanda Fedkiwa, Nicka Fostera i Josa Stama w latach 1996 – 2001 [2, 3, 4, 5, 6]. Obecnie trudno znaleźć film lub grę, w których nie byłyby zastosowane w mniejszym bądź większym stopniu symulacje płynów.

Celem pracy było wykonanie systemu umożliwiającego symulację zachowania płynu, a następnie wykorzystanie wyników symulacji do stworzenia realistycznej animacji puchu śnieżnego wzbitego w powietrze przez przejeżdżającego snowboardzistę, wiatr czy armatkę śnieżną.

### 2. Symulacja płynu

Przepływ płynu w czasie i przestrzeni jest opisany przez nieliniowe równanie

$$\frac{\partial \bar{u}}{\partial t} = \bar{g} - \bar{u} \cdot \nabla \bar{u} + \nu \nabla^2 \bar{u} - \frac{1}{\rho} \nabla p \quad (1)$$

znane jako równanie Naviera-Stokesa. Wektor  $\bar{u}$  jest wektorem prędkości lokalnej płynu,  $\rho$  oznacza gęstość płynu,  $p$  to lokalne ciśnienie,  $\nu$  oznacza kinematyczną lepkość płynu, zaś wektor  $\bar{g}$  oznacza przyspieszenie grawitacyjne, lub bardziej ogólnie, działanie sił zewnętrznych. Występujące po prawej stronie składniki oznaczają kolejno wpływ sił zewnętrznych, adwekcji, dyfuzji i ciśnienia na przyspieszenie.

Równanie Naviera-Stokesa jest silnie nieliniowe i bardzo trudno jest rozwiązać je metodą analityczną. W celu uproszczenia modelu zakładamy, że płyn, którym się zajmujemy, jest nieściśliwy. Gwarantuje to poniższe równanie, znane jako zasada zachowania masy lub równanie ciągłości:

$$\nabla \cdot \bar{u} = 0 \quad (2)$$

Założenie o nieściśliwości nie wprowadza znaczącego błędu, pod warunkiem, że prędkość płynu jest znacznie mniejsza od prędkości rozchodzenia się dźwięku w tym płynie, co dla naszych zastosowań będzie zawsze spełnione.

W celu rozwiązania równań Naviera-Stokesa musimy wprowadzić dyskretyzację czasu i przestrzeni. Dyskretyzacja czasu polega na zastosowaniu stałych przyrostów czasowych przy obliczeniach w kolejnych iteracjach symulacji. Przy czym w celu osiągnięcia dokładniejszych wyników stosujemy kilka iteracji symulacji dla jednej klatki animacji. Dyskretna przestrzeń symulacji to siatka złożona z sześciennych wokseli (wolumen).

Dla każdego wokseli o krawędzi  $\tau$  definiujemy wartości opisujące stan płynu w odpowiadającym miejscu przestrzeni. Wielkości skalarnie (ciśnienie, gęstość, temperatura) są przypisane do geometrycznego środka wokseli, natomiast składowe wektora prędkości płynu są zaczepione w środkach odpowiednich ścian wokseli. Dzięki takiej dyskretyzacji możemy z akceptowalną dokładnością rozwiązać numerycznie kolejne składniki równania 1.

Algorytm symulacji rozkładu prędkości w wolumenie jest następujący:

1. Ustaw warunki brzegowe
2. Wykonaj obliczenia adwekcji
3. Dodaj wpływ sił zewnętrznych
4. Dla każdej iteracji obliczania dyfuzji
  - a. Ustaw warunki brzegowe
  - b. Wykonaj krok obliczający dyfuzję
5. Dopóki pole nie ma zerowej dywergencji lub nie przekroczono ograniczenia iteracji
  - a. Ustaw warunki brzegowe
  - b. Wykonaj krok obliczający ciśnienie
  - c. Dodaj wpływ ciśnienia do pola prędkości

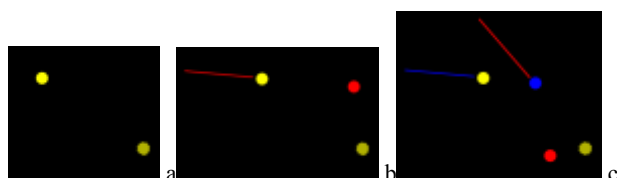
Jedynym krokiem, który nie wymaga ustawiania globalnych warunków brzegowych jest wpływ sił zewnętrznych. Ten krok można wykonać dla każdego woksela lokalnie, bowiem uzyskane wartości zależą jedynie od wartości w tym samym wokselu w poprzednim kroku. Przed innymi krokami symulacji ustawiamy warunki brzegowe. Polega to na przypisaniu wartości na granicach wolumenu. W pierwszej kolejności wykonujemy krok adwekcji. Jest to konieczne, ponieważ adwekcja musi być zastosowana do pola bez dywergencji, które jest wynikiem ostatniego kroku algorytmu. Kolejny krok, czyli dodanie wpływu dyfuzji, jest opcjonalny. Ostatnim i najbardziej skomplikowanym krokiem jest dodanie wpływu ciśnienia.

Operacja dodania adwekcji może spowodować niestabilność symulatora w przypadku, gdy istnieje możliwość, aby cząstka przesunęła się o więcej niż jeden woxsel w pojedynczej iteracji. Można ten warunek zapisać dla  $k$ -tej iteracji, jako:

$$\Delta t |u_{\max}^k| \geq \tau \quad (3)$$

gdzie  $u_{\max}^k$  oznacza maksymalną prędkość płynu w przestrzeni w danej iteracji.

Rozwiązanie zapewniające stabilność zaproponował w 1999 roku Jos Stam [6]. W każdej iteracji, cząstki płynu przesuwają się zgodnie z lokalnymi prędkościami. Do wyznaczenia prędkości płynu w czasie  $t + \Delta t$  wystarczy prześledzić wstecz drogę cząstki od poprzedniej klatki, co obrazuje rysunek 1a:



Rys. 1. Wyznaczanie prędkości w kolejnych krokach symulacji  
Fig. 1. Velocity vector in subsequent simulation steps

Taka operacja spowoduje przesunięcie całego pola prędkości, co dobrze oddaje samounoszenie płynu (adwekcję). Dokładną wartość szukanej prędkości w wybranym punkcie przestrzeni możemy otrzymać, stosując interpolację liniową najbliższych wartości. Zaczepienie wektorów prędkości na ścianach woksela powoduje zmniejszenie błędu oraz ułatwia implementację. Metoda Stama jest stabilna, ponieważ mamy gwarancję, że w wyniku dodania adwekcji, maksymalna prędkość w kroku  $k+1$  nie przekroczy maksymalnej prędkości z kroku  $k$ -tego.

Metoda Stama zastosowana wprost, wprowadza duże błędy wynikające z różnic czasu pomiędzy klatkami. Jeżeli pobierzemy prędkość cząstki poruszającej się po krzywej, otrzymamy pozycję początkową odległą od rzeczywistej (prawy górny punkt na rysunku 1b). W celu rozwiązania tego problemu stosujemy metodę Rungego-Kutty iteracyjnego rozwiązywania równań różniczkowych zwyczajnych. W tym przypadku wystarczyło zastosowanie metody R-K drugiego rzędu:

$$\begin{aligned} \bar{x}_{temp} &= \bar{x}_k - \frac{\Delta t}{2} \bar{u}(\bar{x}_k) \\ \bar{x}_p &= \bar{x}_k - \Delta t \bar{u}(\bar{x}_{temp}) \\ \bar{u}_{t+\Delta t}(\bar{x}_k) &= \bar{u}_t(\bar{x}_p) \end{aligned} \quad (4)$$

Najpierw obliczamy punkt pośredni, cofając się z punktu końcowego o połowę przyrostu czasu. Drugie równanie wykorzystuje prędkość z tego punktu i oblicza położenie przybliżonego punktu początkowego, w którym cząstka znajdowała się w poprzedniej iteracji. Na końcu ustawiamy wartość prędkości w punkcie końcowym ruchu cząstki, pobierając prędkość z poprzedniej iteracji w punkcie początkowym (rysunek 1c). Z punktu pośredniego (w prawej górnej ćwiartce rysunku 1c) pobieramy nową prędkość - pokazaną przez wektor z tego punktu. Najniższy punkt na rysunku to obliczony z metody R-K punkt początkowy. Jak widać, błąd został zmniejszony, lecz nie wyeliminowany. Zastosowanie kolejnych rzędów metody R-K, wprowadzających coraz więcej iteracji, zapewniałoby coraz bliższe rzeczywistemu punkty początkowe, a przez to coraz bardziej realistyczne zachowanie płynu. Jest to jednak rozwiązanie kosztowne obliczeniowo.

Uwzględnienie dyfuzji wprowadza wizualne wygładzenie kształtu płynu, zwłaszcza przy większej liczbie zastosowanych iteracji. W praktyce pominięcie tego kroku nie oznacza braku dyfuzji, ponieważ w wyniku błędów numerycznych i tak zostają wprowadzone pewne drobne wygładzenia. Decyzję, czy stosować dyfuzję trzeba podjąć, oceniając finalne efekty wizualne.

Równanie Poissona opisujące proces dyfuzji:

$$\nabla^2 v = f \quad (5)$$

może być rozwiązane metodą Jacobiego:

$$v_{i,j,k}^{t+1} = \frac{v_{i-1,j,k}^t + v_{i+1,j,k}^t + v_{i,j-1,k}^t + v_{i,j+1,k}^t + v_{i,j,k-1}^t + v_{i,j,k+1}^t + \alpha b_{i,j,k}^t}{\beta} \quad (6)$$

gdzie  $v$  jest pewną wartością dla której stosujemy metodę Jacobiego przy przejściu z iteracji  $t$  do  $t+1$ . Za  $v$  oraz  $b$  podstawiamy wektory prędkości z odpowiednich wokseli. Ponieważ  $\alpha = \tau^2 / v\Delta t$ , zaś  $\beta = \alpha + 6$ , zatem w każdej kolejnej iteracji, dla każdego woksela wykonujemy następującą operację:

$$u_{i,j,k}^{-t+1} = \frac{v\Delta t (\bar{u}_{i-1,j,k}^{-t} + \bar{u}_{i+1,j,k}^{-t} + \bar{u}_{i,j-1,k}^{-t} + \bar{u}_{i,j+1,k}^{-t} + \bar{u}_{i,j,k-1}^{-t} + \bar{u}_{i,j,k+1}^{-t}) + \bar{u}_{i,j,k}^{-t} \tau^2}{\tau^2 + 6v\Delta t} \quad (7)$$

Uwzględnienie wpływu ciśnienia płynu i jednocześnie zapewnienie zerowej dywergencji pola prędkości jest najtrudniejszym zadaniem przy budowaniu symulatora. Jednocześnie, ponieważ obliczenia te wymagają wielu iteracji, można sterować czasem symulacji przez zmianę ograniczenia górnego na ich liczbę. Więcej iteracji daje oczywiście mniejszy błąd w wyniku końcowym.

Twierdzenie Helmholtza mówi, że pole wektorowe  $F$  może zostać przedstawione w postaci sumy gradientu pewnej funkcji skalarniej  $\phi$  oraz pola wektorowego  $A$  o zerowej dywergencji:

$$\begin{aligned} F &= \nabla \phi + A \\ \nabla \cdot A &= 0 \end{aligned} \quad (8)$$

Powyższe twierdzenie możemy zastosować w celu uwzględnienia ciśnienia w modelu płynu. W tym celu wystarczy zauważyć, że po dodaniu wpływu sił zewnętrznych, adwekcji oraz dyfuzji do pola wektorowego prędkości otrzymujemy nowe pole o niezerowej dywergencji. Oznaczmy wektory z tego pola przez  $\bar{w}_{i,j,k}$ . Możemy otrzymać nowe pole wektorowe bez dywergencji, stosując twierdzenie Helmholtza w następujący sposób:

$$\bar{w}_{i,j,k} = \nabla p + \bar{u}_{i,j,k} \quad (9)$$

Po zastosowaniu operatora Laplace'a do obu stron równania i uwzględnieniu zerowej dywergencji otrzymamy:

$$\nabla \cdot \bar{w}_{i,j,k} = \nabla^2 p \quad (10)$$

Rozwiązanie można znaleźć metodą Jacobiego przez podstawienie w równaniu 6 za  $v$  wartości ciśnienia, za  $b$  dywergencję pola  $W$ . Dodatkowo  $\alpha = \tau^2$ , zaś  $\beta = 6$ , zatem metoda ta prezentuje się następująco:

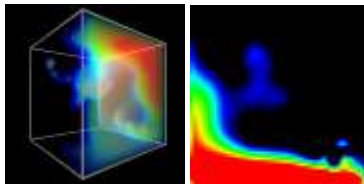
$$p_{i,j,k}^{t+1} = \frac{p_{i-1,j,k}^t + p_{i+1,j,k}^t + p_{i,j-1,k}^t + p_{i,j+1,k}^t + p_{i,j,k-1}^t + p_{i,j,k+1}^t + \nabla \cdot \bar{u}_{i,j,k}^t \tau^2}{6} \quad (11)$$

Po każdej iteracji uaktualniającej ciśnienie we wszystkich woksela, zgodnie z twierdzeniem Helmholtza zmieniamy pole wektorowe prędkości, odejmując od prędkości w woksela gradient ciśnienia w ich otoczeniu:

$$\bar{u}_{i,j,k}^{t+1} = \bar{u}_{i,j,k}^t - \nabla p_{i,j,k} \quad (12)$$

Następnie sprawdzamy jaka jest obecna dywergencja pola i w zależności od potrzeb wykonujemy kolejną iterację.

Przykładowy rozkład ciśnienia płynu przedstawia rysunek 2. Po lewej stronie widoczny jest rozkład w trzech wymiarach, po prawej ten sam rozkład w rzucie ortogonalnym. Ciepłsza barwa oznacza obszar o wyższym ciśnieniu.



Rys. 2. Przykładowy rozkład ciśnienia płynu  
Fig. 2. Sample distribution of fluid pressure

Opisane dotąd procedury prowadzą do określenia pola prędkości, czyli sparametryzowania ośrodka, w którym może rozchodzić się płyn. Symulacja konkretnej substancji wymaga nowego pola skalarnego (lub kilku pól) przechowywanego gęstość płynu. Zachowanie gęstości można opisać równaniem adwekcyjno-dyfuzyjnym [2]:

$$\frac{\partial \rho}{\partial t} = -\bar{u} \nabla \rho + \gamma \nabla^2 \rho - \alpha \rho + s \quad (13)$$

Kolejne składniki po prawej stronie równania oznaczają wpływ adwekcji, dyfuzji, rozpraszania i źródłowości.

Wpływ adwekcji i dyfuzji jest analogiczny jak w przypadku pola prędkości. Trzeba jedynie pamiętać, że należy w inny sposób dokonać interpolacji, ponieważ teraz wartości są przechowywane w środkach woksela. Rozpraszanie polega na zaniku substancji w czasie, przy czym współczynnik  $\alpha$  służy do sterowania prędkością tego procesu:

$$\frac{\rho^{t+\Delta t} - \rho^t}{\Delta t} = -\alpha \rho^t \quad (14)$$

Należy jednak uważać, aby  $\alpha \Delta t \in (0,1)$ .

Mniej poprawna fizycznie, lecz nie wymagająca stosowania się do powyższego ograniczenia, jest następująca metoda:

$$\rho^{t+\Delta t} = \frac{\rho^t}{(1 + \alpha \Delta t)} \quad (15)$$

w której wystarczy zapewnić, aby  $\alpha \geq 0$ . W wykonanych testach obie metody dawały bardzo podobne wizualnie wyniki. Pierwsza metoda może powodować pewne problemy w przypadku, gdy nie znamy z góry wartości  $\Delta t$ , co może się zdarzyć, jeśli nasz symulator ma działać w czasie rzeczywistym.

Ostatnim krokiem jest uwzględnienie źródłowości, można ją interpretować jako uzupełnianie brakującej gęstości w wolumenie lub usuwanie gęstości nadmiarowej. Odpowiedzialne za to są źródła (dodatnie  $s$ ) i ujścia (ujemne  $s$ ) umieszczone w wolumenie.

W celu określenia zachowania płynu na obrzeżach wolumenu, stosujemy specjalne warunki brzegowe. Jeśli przestrzeń jest otwarta, to zakładamy, że prędkość na zewnątrz siatki jest zerowa, a gęstość i ciśnienie takie jak w woksela znajdującym się najbliższym miejscu, z którego je pobieramy. Natomiast w przypadku wolumenu ograniczonego stałymi ścianami (akwarium), gęstość i ciśnienie poza siatką są zerowe, a wektor prędkości na zewnątrz siatki jest taki, jak wektor z woksela, który znajduje się najbliższym danego punktu, ale o przeciwnym kierunku. Przy czym można tutaj mnożyć przez  $-1$  tylko wybrane (w zależności od odległości danego punktu od ściany akwarium) składowe wektora prędkości, a pozostałym przypisać zero, lub też cały wektor mnożyć przez  $-1$ , co powoduje większe turbulencje na granicach domeny.

Tab. 1. Warunki brzegowe dla wolumenu  
Tab. 1. Volume border conditions

	Akwarium	Otwarta przestrzeń
Prędkość na zewnątrz	$-u$	$(0,0,0)$
Ciśnienie na zewnątrz	0	$p$
Gęstość na zewnątrz	0	$\rho$

W naszym modelu można w prosty sposób dodać przeszkody na drodze płynu. Wymaga to zmiany wartości prędkości i gęstości w miejscach, w których znajduje się stała przeszkoda. Prędkość przyjmuje wartość prędkości przeszkody (w przypadku szczególnym, gdy przeszkoda nie porusza się, będzie to wektor zerowy), a gęstość wartość zerową.

W wyniku symulacji płynu na dyskretnej siatce woksela oraz interpolacji wartości pomiędzy woksela dochodzi nie tylko do wygładzenia wizualnego (przypominającego dyfuzję), wygładzane są również zawirowania małej skali powstające w płynie. Aby temu zaradzić, stosuje się algorytm Vorticity Confinement, którego przebieg ilustrują poniższe wzory:

$$\begin{aligned} \bar{\omega} &= \nabla \times \bar{u} = \left( \frac{\partial w}{\partial y} - \frac{\partial v}{\partial z}, \frac{\partial u}{\partial z} - \frac{\partial w}{\partial x}, \frac{\partial v}{\partial x} - \frac{\partial u}{\partial y} \right) \\ \bar{\Psi} &= \frac{\nabla \bar{\omega}}{|\nabla \bar{\omega}|} \\ \bar{f}_v &= \varepsilon \tau (\bar{\Psi} \times \bar{\omega}) \end{aligned} \quad (16)$$

Operator  $\nabla \times \bar{u}$  to tak zwany operator rotacji, który można zapisać również jako  $rot u$ . Wynikowy wektor  $\bar{\omega}$  jest zawsze prostopadły do płaszczyzny rotacji, a jego długość określa siłę rotacji. Znormalizowany gradient  $\bar{\Psi}$  wskazuje kierunek wzrostu rotacji. Przeskalowany iloczyn tych dwóch wektorów opisuje siłę  $\bar{f}_v$  wzmacniającą rotację w danym miejscu przestrzeni. Można ją wykorzystać jako siłę zewnętrzną działającą na płyn. Algorytm ten, choć nie jest poprawny fizycznie, powoduje znaczącą poprawę wizualną. W rzeczywistości jest to algorytm, który ma za zadanie maskowanie błędów numerycznych wynikających z interpolacji w dyskretnej siatce woksela.

Współczynnik  $\varepsilon$  umożliwia sterowanie turbulentnością przepływu, jednak należy używać go rozważnie. Przy zbyt dużej wartości tego współczynnika, istnieje możliwość destabilizacji modelu.

### 3. Rendering

Siatki woksela opisujące dystrybucję gęstości płynu dla każdego kroku symulacji są danymi wejściowymi dla systemu renderingu. Pojedynczą klatkę animacji uzyskaliśmy, wykorzystując algorytm śledzenia promieni w przestrzeni woksela. Skuteczną metodą okazało się zastosowanie otwartego oprogramowania

Blender3D. Zaletą tego rozwiązania jest prosta integracja naszego symulatora płynu z rendererem, a także możliwość tworzenia scen trójwymiarowych otaczających symulowane zjawiska.

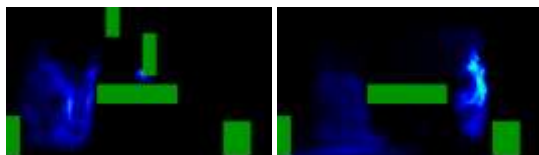
Uzyskanie realistycznych efektów wizualizacji wymaga odpowiedniego zdefiniowania w rendererze własności optycznych materiałów (śnieg), źródeł światła i ich wzajemnej interakcji. Przy odrobinie wprawy można uzyskać akceptowalne efekty, korzystając z standardowych mechanizmów programu Blender3D.

## 4. Wyniki

### Animacja „Paśnik”

Animacja ta powstała na podstawie fragmentu filmu „In Deep” o tematyce górskiej, w którym narciarz wyskakuje z muldy przed paśnikiem dla zwierząt, uderza o niego rękoma i ląduje po drugiej stronie, wzbudzając dwukrotnie tuman puchu śnieżnego.

Symulacja na siatce o wymiarach 152x86x76 wokseli trwała 80 klatek. W przestrzeni występują trzy nieruchome przeszkody reprezentujące dwie muldy (jedną, z której obiekt się wybija, oraz drugą, na której ląduje) oraz daszek paśnika. Dwie ruchome przeszkody poruszające się ruchem prostoliniowym symulują obiekt wzbijający puch i pojawiają się w odpowiednich klatkach animacji. Rysunek 3 przedstawia rozkłady gęstości w 38 i 50 klatce animacji.



Rys. 3. Rozkład gęstości w klatkach animacji. Prostokątne obiekty reprezentują przeszkody

Fig. 3. Density distribution in selected frames. Rectangle objects represent obstacles

Na cztery klatki przed uruchomieniem każdej z ruchomych przeszkód, w miejscu jej pojawienia się powstaje nieruchome źródło siły. Symuluje to ruch powietrza wokół narciarza. W momencie pojawienia się przeszkody umieszczane jest pod nią źródło gęstości o małych rozmiarach, które znika po czasie 0.3 s, czyli po około 7 klatkach animacji (rysunek 4).

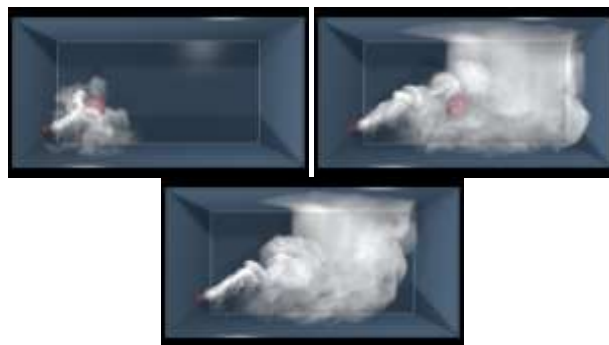


Rys. 4. Wybrane klatki animacji „Paśnik”

Fig. 4. Selected frames of „Pasture” animation

### Animacja „Armatka śnieżna”

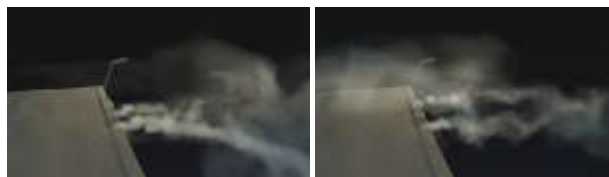
Animacja pokazuje w jaki sposób przeszkody wpływają na poruszanie się płynu. Armatka śnieżna (umieszczona po lewej stronie obrazu) to jednostajne źródło gęstości o niewielkich rozmiarach. Przeszkodami są stałe ściany dużego akwarium, w którym odbywa się symulacja i kulka poruszająca się wewnątrz. Symulacja została przeprowadzona na siatce o wymiarach 200x100x100 wokseli (rysunek 5).



Rys. 5. Wybrane klatki animacji „Armatka śnieżna”  
Fig. 5. Selected frames of „Snow cannon” animation

### Animacja „Dach”

Ciekawym zjawiskiem jest też zdmuchiwanie śniegu z dachu. Specyficzny efekt powstaje zwłaszcza w miejscach, gdzie są umieszczone rowki w pokrywie dachowej. W tej symulacji opisano źródła siły i gęstości wraz z podaniem klatek, w których mają się pojawić. Siatka wokselowa ma rozmiary 140x60x70. Pojedyncza, duża przeszkoda symuluje budynek, zastosowane są warunki brzegowe dla otwartej przestrzeni (rysunek 6).



Rys. 6. Wybrane klatki animacji „Dach”  
Fig. 6. Selected frames of „Roof” animation

Tabela 2 przedstawia średni czas symulacji pojedynczej klatki oraz wymagania pamięciowe w zależności od rozmiarów siatki, ograniczenia na liczbę iteracji symulatora ciśnienia oraz od tego, czy vorticity confinement (kolumna VC) jest włączony. Prezentowany czas jest uśrednioną wartością z 24 klatek, co odpowiada jednej sekundzie animacji końcowej. Testy zostały przeprowadzone na procesorze Intel Core i3 350M 2.27GHz, przy 4 GB pamięci RAM w 64-bitowym systemie Microsoft Windows 7.

Istotny wpływ na wymagania pamięciowe ma tylko rozmiar siatki. Wzrost czasu jest bardzo podobny przy podwojeniu rozmiarów siatki i ograniczeniu liczby iteracji. Oznacza to, że obliczenia dla ciśnienia są najbardziej kosztowną częścią symulacji.

Tab. 2. Parametry symulacji pojedynczej klatki animacji  
Tab. 2. Simulation parameters for a single animation frame

Rozmiar siatki	Iteracje	VC	Czas, s	Pamięć, MB
128x128x128	100	Nie	26,524	80
128x128x128	100	Tak	27,124	80
128x128x256	100	Nie	55,314	160
128x128x256	100	Tak	56,739	160
128x128x128	200	Nie	51,914	80
128x128x128	200	Tak	52,872	80

## 5. Podsumowanie

W pracy zaprezentowano metodę implementacji symulatora płynów do stworzenia animacji przedstawiających wyrzucony w powietrze puch śnieżny.

W ramach podsumowania warto wspomnieć o kilku problemach związanych z animacją płynów. Pierwszym są ogromne wymagania pamięciowe. Większe rozmiary siatek symulacji i precyzja potrzebnych obliczeń zmiennoprzecinkowych powodują skokowy wzrost rozmiarów pamięci niezbędnych do przechowywania danych obliczeniowych dla każdego kroku symulacji.

Kolejnym problemem jest złożoność obliczeniowa rzędu  $O(n^3)$ , która powoduje ogromny nakład czasowy na wykonanie symulacji.

Problemem zupełnie innej natury jest określenie, czy wyniki są wystarczająco realistyczne. Jakość animacji możemy określać jedynie intuicyjnie, w oparciu o nasze doświadczenie. Pojawia się tu również problem braku rozwiązania optymalnego - nigdy nie można stwierdzić, że animacji nie da się zrobić "lepiej", czyli bardziej realistycznie.

## 6. Literatura

[1] Bridson R.: Fluid Simulation for Computer Graphics. A K Peters, 2008.

[2] Fedkiw R., Stam J., Jensen H.W.: Visual Simulation of Smoke. Computer Graphics Proceedings Annual Conference Series, Siggraph'01, pp.15-22, 2001.

[3] Foster N., Metaxas D.: Realistic Animation of Liquids. Graphical Models and Image Proc., vol.58(5), pp.471-483, 1996.

[4] Foster N., Metaxas D.: Modeling the Motion of a Hot, Turbulent Gas. Computer Graphics Proceedings Annual Conference Series, Siggraph'97, pp.181-188, 1997.

[5] Foster N., Fedkiw R.: Practical Animation of Liquids. Computer Graphics Proceedings Annual Conference Series, Siggraph'01, pp.23-30, 2001.

[6] Stam J.: Stable Fluids. Computer Graphics Proceedings Annual Conference Series, Siggraph'99, pp.121-129, 1999.

*otrzymano / received: 06.02.2014*

*przyjęto do druku / accepted: 01.04.2014*

*artykuł recenzowany / revised paper*

## INFORMACJE

### Informacje dla Autorów

Redakcja przyjmuje do publikacji tylko prace oryginalne, nie publikowane wcześniej w innych czasopismach. Redakcja nie zwraca materiałów nie zamówionych oraz zastrzega sobie prawo redagowania i skracania tekstów oraz streszczeń.

Artykuły naukowe publikowane w czasopiśmie PAK są formatowane jednolicie zgodnie z ustaloną formatką zamieszczoną na stronie redakcyjnej [www.pak.info.pl](http://www.pak.info.pl). Dlatego artykuły przekazywane redakcji należy przygotowywać w edytorze Microsoft Word 2003 (w formacie DOC) z zachowaniem:

- wielkości czcionek,
- odstępów między wierszami tekstu,
- odstępów przed i po rysunkach, wzorach i tabelach,
- oznaczeń we wzorach, tabelach i na rysunkach zgodnych z oznaczeniami w tekście,
- układu poszczególnych elementów na stronie.

Osobno należy przygotować w pliku w formacie DOC notki biograficzne autorów o objętości nie przekraczającej 450 znaków, zawierające podstawowe dane charakteryzujące działalność naukową, tytuły naukowe i zawodowe, miejsce pracy i zajmowane stanowiska, informacje o uprawianej dziedzinie, adres e-mail oraz aktualne zdjęcie autora o rozmiarze 3,8 x 2,7 cm zapisane w skali odcieni szarości lub dołączone w osobnym pliku (w formacie TIF).

Wszystkie materiały:

- artykuł (w formacie DOC),
- notki biograficzne autorów (w formacie DOC),
- zdjęcia i rysunki (w formacie TIF lub CDR),

prosimy przysłać w formie plików oraz dodatkowo jako wydruki na białym papierze (lub w formacie PDF) na adres e-mail: [wydawnictwo@pak.info.pl](mailto:wydawnictwo@pak.info.pl) lub pocztą zwykłą, na adres: Redakcja Czasopisma Pomiary Automatyka Kontrola, Asystent Redaktora Naczelnego mgr Agnieszka Skórkowska, ul. Akademicka 10, p.21A, 44-100 Gliwice.

Wszystkie artykuły naukowe są dopuszczane do publikacji w czasopiśmie PAK po otrzymaniu pozytywnej recenzji. Autorzy materiałów nadesłanych do publikacji są odpowiedzialni za przestrzeganie prawa autorskiego. Zarówno treść pracy, jak i wykorzystane w niej ilustracje oraz tabele powinny stanowić dorobek własny Autora lub muszą być opisane zgodnie z zasadami cytowania, z powołaniem się na źródło cytatu.

Przedrukowywanie materiałów lub ich fragmentów wymaga pisemnej zgody redakcji. Redakcja ma prawo do korzystania z utworu, rozporządzania nim i udostępniania dowolną techniką, w tym też elektroniczną oraz ma prawo do rozpowszechniania go dowolnymi kanałami dystrybucyjnymi.