

# Hydro-Dynamica

## Rozwiązanie Numeryczne Równań Przepływu Cieczy Nieściśliwej.

Maciej Matyka

Fizyka Komputerowa V BIS  
Uniwersytet Wrocławski  
Wydział Fizyki i Astronomii

maq@panoramix.ift.uni.wroc.pl  
<http://panoramix.ift.uni.wroc.pl/~maq>

28 lutego 2004 roku

### Streszczenie

Program Hydro-Dynamica rozwiązuje problem przepływu cieczy dla dowolnej geometrii przeszkód oraz źródeł cieczy. Niniejsza dokumentacja opisuje podstawy fizyczne i numeryczne zastosowanego modelu oraz metody rozwiązania. Do pełnego rozwiązania równań nieliniowych przepływu cieczy nieściśliwej użyta została metoda SIMPLE, która mimo nazwy okazuje się wcale nie taka 'prosta'.

### 1 Wstęp

Założeniem programu, który niniejsza dokumentacja będzie opisywać, było, by dać użytkownikowi narzędzie do rozwiązania w miarę dowolnego problemu dwuwymiarowego przepływu cieczy nieściśliwej.

Wybór modelu opisującego zachowanie cieczy oraz metody rozwiązania potrzebnych równań nie był prosty. Istnieje bowiem kilka różnych podejść m. in. metoda cząstek znaczonych [8, 7], w której Harlow i Welch jako pierwsi podjęli się numerycznego rozwiązania problemów hydrodynamiki nieściśliwej przez bezpośrednie rozwiązanie równań Naviera-Stokesa oraz użycie siatki obliczeniowej Eulera. Wprowadzili oni do modelu bezmasowe pseudo- cząstki, które pozwalają w prosty

sposób śledzić ruch powierzchni swobodnej cieczy<sup>1</sup>. Innym podejściem do rozwiązania równań Naviera-Stokesa jest podejście Lagrange'a [2], które sprowadza się do śledzenia trajektorii każdej cząsteczki cieczy z osobna.

Dla potrzeb programu "HydroDynamica" wybrałem metodę SIMPLE, której opis w polskiej literaturze występuje m.in. w [2], a dokładniejszy opis algorytmu znaleźć można w angielskojęzycznych pozycjach [1, 4, 5, 6].

Nazwa metody - 'SIMPLE' - jest skrótem od 'Semi Implicit Method for Pressure Linked Equations', co na potrzeby niniejszej dokumentacji przetłumaczmy jako 'Pseudo Niejawną Metodę dla Równań Łączonych przez Ciśnienie<sup>2</sup>'. Przy okazji - będę niezmiernie wdzięczny za

<sup>1</sup>Patrz również program 'Fluid'.

<sup>2</sup>SIMPLE = PNMRLC? :)

wszelkie uwagi co do ww tłumaczenia. Metoda SIMPLE wybrana została ze względu na kilka cech, które okazały się być bardzo użyteczne w programie o jakim mowa była na początku ww wstępu: wystarczająca dokładność rozwiązania, użycie metody siatek rozbieżnych (ułatwienie w implementacji), stabilność i dość bezpośrednia oraz intuicyjna konstrukcja warunków brzegowych.

## 2 Równania Naviera-Stokesa

Dla potrzeb programu zapiszmy dwuwymiarowe równania Naviera-Stokesa w postaci bezwymiarowej: <sup>3</sup>:

$$\frac{\partial \vec{u}}{\partial t} = -(\vec{u} \nabla) \vec{u} - \nabla \varphi + \frac{1}{Re} \nabla^2 \vec{u} \quad (1)$$

$$D = \nabla \vec{u} = 0 \quad (2)$$

Gdzie równanie (2) jest równaniem ciągłości i musi być spełnione w trakcie całego czasu symulacji.

## 3 Metody oparte na zmiennych prymitywnych

Algorytm SIMPLE, który będziemy opisywać wymaga użycia formuły równań NS opartych na zmiennych prymitywnych. Oznacza to tyle, że używać będziemy formy równań NS ze zmiennymi  $u, v, p$ , gdzie  $\mathbf{u} = (u, v)$  jest wektorem prędkości, a  $p$  oznacza ciśnienie. Możemy przepisać równania NS w formie różniczkowej dla obu składowych prędkości z osobna:

$$\frac{\partial u}{\partial t} = -\frac{\partial u^2}{\partial x} + \frac{\partial uv}{\partial y} - \frac{\partial p}{\partial x} + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (3)$$

$$\frac{\partial v}{\partial t} = -\frac{\partial v^2}{\partial y} + \frac{\partial uv}{\partial x} - \frac{\partial p}{\partial y} + \frac{1}{Re} \left( \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) \quad (4)$$

Równanie ciągłości przepiszmy w formie:

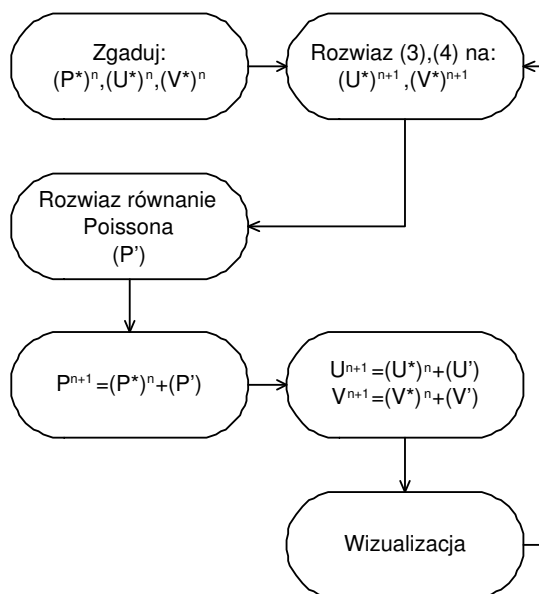
$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (5)$$

Oba powyższe równania użyte zostaną jako punkt wyjścia do rozwiązania metodą SIMPLE.

### 3.1 Algorytm SIMPLE

Szkic algorytmu SIMPLE rozwiązania równań NS przedstawiony został na rysunku (1).

<sup>3</sup>Rozpatrujemy przepływy bez sił zewnętrznych, np. grawitacji.



Rysunek 1: Szkic algorytmu SIMPLE

Najpierw zgadujemy początkowe wartości pola ciśnienia<sup>4</sup>  $(P^*)^n$  i ustalamy (odgadujemy) pola prędkości -  $(U^*)^n, (V^*)^n$ . Wtedy równania (3) i (4) są rozwiązane dla uzyskania wartości  $(U^*)^{n+1}, (V^*)^{n+1}$ . Po tych krokach musimy rozwiązać równanie korekty ciśnienia:

$$\nabla^2 p' = \frac{1}{\Delta t} (\nabla \cdot V) \quad (6)$$

Po tym, używamy prostych relacji, aby uzyskać poprawione wartości pól ciśnienia i prędkości. Przy rozwiązywaniu problemów stacjonarnych możemy też sprawdzić w tym miejscu czy rozwiązanie jest zbieżne.

### 3.2 Metody numeryczne w SIMPLE

#### 3.2.1 Siatka Rozbieżna

Dla dyskretyzacji równań różniczkowych NS użyjemy siatki rozbieżnej. Na obrazku (2) została przedstawiona siatka rozbieżna prostokątna. Zmienne prymitywne zostały umiejscowione w różnych punktach siatki. Mówiąc w różnych punktach, mamy na myśli, że ciśnienie, prędkość pozioma i prędkość pionowa NIE znajdują się w tym samym punkcie komórki siatki. Stąd tytułowa "rozbieżność" siatki obliczeniowej. W punktach całkowitych  $i, j$  siatki umiejscowione zostały wartości ciśnienia  $P$ , w punktach połówkowych poziomo -  $i + 0.5, j$

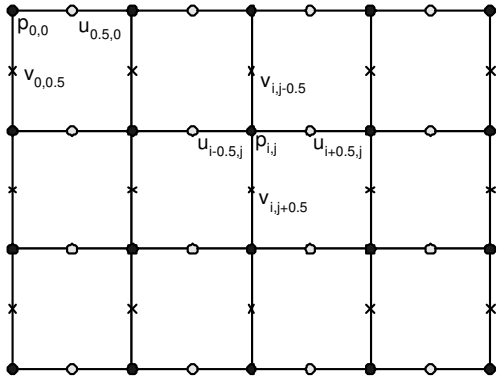
<sup>4</sup>Indeksy oznaczają pojedynczy krok obliczeniowy, gdzie 'n+1' oznacza krok bieżący.

Różniczka	Schemat Różnicowy	Typ, dokładność
$\frac{\partial u}{\partial t}$	$\frac{u^{n+1}-u^n}{\Delta t}$	"w przód", $O(h)$
$\frac{\partial^2 u}{\partial x^2}$	$\frac{u_{i+1,j}-2u_{i,j}+u_{i-1,j}}{(\Delta x)^2}$	"centralny", $O(h^2)$
$\frac{\partial u^2}{\partial x}$	$\frac{u_{i+1,j}^2-u_{i-1,j}^2}{(2 \cdot \Delta x)}$	"centralny", $O(h^2)$
$\frac{\partial p}{\partial x}$	$\frac{p_{i+1,j}-p_{i,j}}{(\Delta x)}$	"w przód", $O(h)$
$\frac{\partial p}{\partial y}$	$\frac{p_{i,j+1}-p_{i,j}}{(\Delta x)}$	"w przód", $O(h)$

Tabela 1: Schematy dyskretyzacji użyte w algorytmie SIMPLE.

oraz w punktach połówkowych pionowo -  $i, j+0.5$  położone zostały wartości prędkości odpowiednio  $u$  oraz  $v$ .

Tak zdefiniowana siatka rozbieżna pozwala na użycie prostych dyskretyzacji i osiągnięcie dokładności drugiego rzędu [8, 4, 5, 6].



Rysunek 2: Siatka rozbieżna: pełne kółka  $P$ , kółka puste  $U$  prędkość  $x$ , krzyż  $V$  składowa  $y$ -prędkości.

### 3.2.2 Dyskretyzacja

Pierwszym krokiem przy rozwiązaniu numerycznym równań NS będzie przeprowadzenie dyskretyzacji na siatce obliczeniowej.

W tabeli 1 przedstawione zostały wszystkie schematy użyte do dyskretyzacji równań dynamiki. Używając zaprezentowanych schematów wyprowadzić możemy

numeryczne schematy obliczeniowe [1]. Równania (3) , (4) zdyskretyzowane na siatce rozbieżnej zapisać możemy w formie<sup>5</sup> <sup>6</sup>:

$$u_{i+0.5,j}^{n+1} = u_{i+0.5,j}^n + \Delta t \cdot (A - (\Delta x)^{-1}(p_{i+1,j} - p_{i,j})) \quad (7)$$

$$v_{i,j+0.5}^{n+1} = v_{i,j+0.5}^n + \Delta t \cdot (B - (\Delta y)^{-1}(p_{i,j+1} - p_{i,j})) \quad (8)$$

gdzie  $A$  i  $B$  zostały zdefiniowane jako:

$$A = -a_1 + (Re)^{-1} \cdot (a_3 + a_4) \quad (9)$$

$$B = -b_1 + (Re)^{-1} \cdot (b_3 + b_4) \quad (10)$$

odpowiednio zapisujemy też:

$$a_1 = -\frac{(u^2)_{i+1.5,j}^n - (u^2)_{i-0.5,j}^n}{2 \cdot \Delta x} - \frac{(u\dot{v})_{i+0.5,j+1}^n - (u\dot{v})_{i+0.5,j-1}^n}{2 \cdot \Delta y} \quad (11)$$

$$b_1 = -\frac{(v^2)_{i,j+1.5}^n - (v^2)_{i,j-0.5}^n}{2 \cdot \Delta y} - \frac{(v\dot{u})_{i+1,j+0.5}^n - (v\dot{u})_{i-1,j+0.5}^n}{2 \cdot \Delta x} \quad (12)$$

$$(a_3) = \frac{u_{i+1.5,j}^n - 2 \cdot u_{i+0.5,j}^n + u_{i-0.5,j}^n}{(\Delta x)^2} \quad (13)$$

$$(a_4) = \frac{u_{i+0.5,j+1}^n - 2 \cdot u_{i+0.5,j}^n + u_{i+0.5,j-1}^n}{(\Delta y)^2} \quad (14)$$

$$(b_3) = \frac{v_{i,j+1.5}^n - 2 \cdot v_{i,j+0.5}^n + v_{i,j-0.5}^n}{(\Delta y)^2} \quad (15)$$

$$(b_4) = \frac{v_{i+1,j+0.5}^n - 2 \cdot v_{i,j+0.5}^n + v_{i-1,j+0.5}^n}{(\Delta x)^2} \quad (16)$$

Teraz mamy zapisane prawie wszystko. Pozostało zdefiniować prędkości "z kropką":

$$\dot{u} = 0.5 \cdot (u_{i-0.5,j} + u_{i+0.5,j+1}) \quad (17)$$

$$\dot{u} = 0.5 \cdot (u_{i+0.5,j} + u_{i+0.5,j+1}) \quad (18)$$

$$\dot{v} = 0.5 \cdot (v_{i,j+0.5} + v_{i+1,j+0.5}) \quad (19)$$

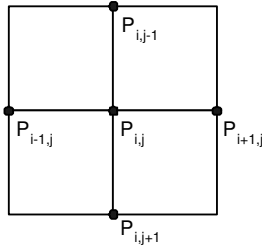
$$\dot{v} = 0.5 \cdot (v_{i,j-0.5} + v_{i+1,j-0.5}) \quad (20)$$

<sup>5</sup>W książce [1] należy uważać na błędy drukarskie w wyprowadzonych schematach numerycznych.

<sup>6</sup>W ogólności przedstawione zostały tylko ogólne schematy całkowania równań, dla potrzeb algorytmu SIMPLE należy oczywiście uzupełnić je o wszelkie indeksy ' lub \*.

### 3.2.3 Równanie Poissona

Dla równania użyjemy prostej metody iteracyjnej wyprowadzonej wprost ze schematów numerycznych dyskretyzacji. Na obrazku (3) zostały zaznaczone punkty użyte do iteracji dla każdego punktu  $(i, j)$  na siatce.



Rysunek 3: Punkty na siatce różnicowej użyte do rozwiązania iteracyjnego równania Poissona.

Użyłem schematu 4-ro punktowego dla przybliżenia operatora Laplace'a. Bezpośrednio z [1] wyrażenie na pojedynczy krok rozwiązania równania Poissona zapisujemy jako:

$$p'_{i,j} = -a^{-1}(b \cdot (p'_{i+1,j} + p'_{i-1,j}) + c \cdot (p'_{i,j+1} + p'_{i,j-1}) + d) \quad (21)$$

gdzie:

$$a = 2\Delta t \left[ \frac{1}{\Delta x^2} + \frac{1}{\Delta y^2} \right] \quad (22)$$

$$b = -\frac{\Delta t}{\Delta x^2} \quad (23)$$

$$c = -\frac{\Delta t}{\Delta y^2} \quad (24)$$

$$d = \frac{1}{\Delta x} [u_{i+0.5,j} - u_{i-0.5,j}] + \frac{1}{\Delta y} [v_{i,j+0.5} - v_{i,j-0.5}] \quad (25)$$

Używamy równania (21) dla wszystkich punktów wewnętrznych na siatce. Następnie sprawdzamy czy otrzymane rozwiązanie zbiegło odpowiednio blisko do wartości dokładnej (inaczej mówiąc sprawdzamy największą różnicę między starym a nowym polem ciśnienia). Jeśli różnica ta jest mniejsza od pewnego skończonego  $\epsilon > 0$ , rozwiązanie uznajemy za dostatecznie dokładne.

## 4 Program Hydro-Dynamica

Program Hydro-Dynamica rozwiązuje równania Naviera-Stokesa na siatce prostokątnej, dwuwymiarowej przy zadanych warunkach brzegowych oraz rozkładu źródeł cieczy. Jak wspomniałem we wstępie do dokumentacji, założeniem było, by użytkownik był w stanie w prosty sposób definiować problem hydrodynamiczny do rozwiązania.

### 4.1 Interfejs użytkownika

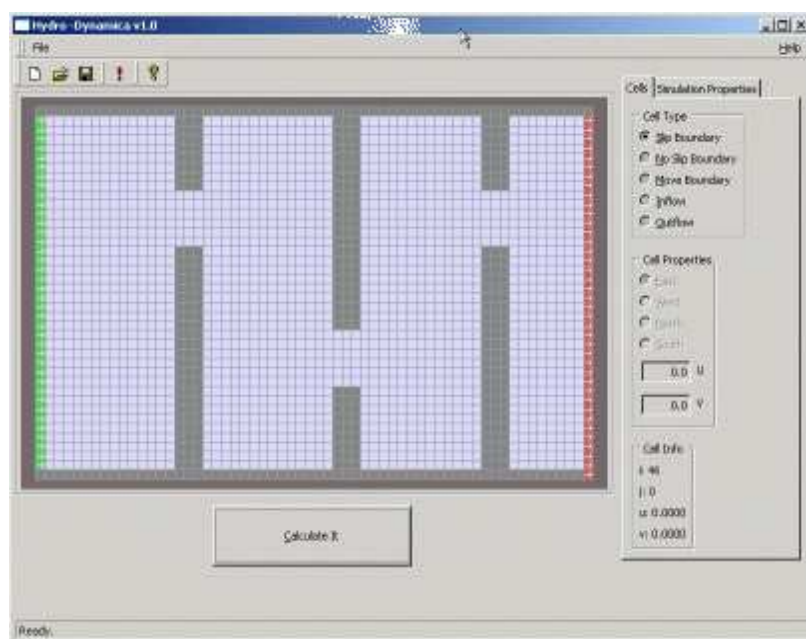
Obsługa programu sprowadza się do zaprojektowania siatki obliczeniowej. Projekt rozpoczynamy od zdefiniowania ogólnych właściwości siatki. Następnie przy pomocy myszki oznaczamy odpowiednie komórki siatki jako komórki wyłączone cieczą, komórki wpływowe / wypływowe lub komórki brzegowe.

Program pozwala użytkownikowi na zdefiniowanie sposobu wizualizacji oraz rodzaju pliku wynikowego (oddzielne klatki lub animacja).

Interfejs użytkownika podzielony został na dwa oddzielne okna.

#### 4.1.1 Okno Główne Programu

Okno główne programu podzielone zostało na dwie części (rysunek 4).



Rysunek 4: Okno główne programu.

Po stronie lewej widoczny jest prostokątny obszar z naszkicowaną siatką obliczeniową. W obszarze tym przy pomocy myszki zaznaczać możemy poszczególne komórki siatki obliczeniowej i zmieniać ich rodzaj. Rodzaj komórki wybieramy z części prawej okna (zakładka "Cells"). Do wyboru mamy następujące rodzaje komórek<sup>7</sup>:

- "Slip Boundary" - komórki brzegowe z warunkiem pełnego poślizgu cieczy
- "No Slip Boundary" - komórki brzegowe z warunkiem braku poślizgu cieczy
- "Move Boundary" - komórki brzegowe z ustaloną prędkością przesuwu
- "Inflow" - komórki wpływu cieczy (z możliwością określenia kierunku wpływu)

<sup>7</sup>Opis bazuje na wersji angielskojęzycznej programu, przy czym bezpośrednio odnosi się również do wersji polskiej.

- "Outflow" - komórki wypływu cieczy

Dla komórek brzegowych poruszających się oraz dla komórek wpływowych można również określić prędkość domyślną w komórce.

Przed przystąpieniem do definiowania własnych układów przeszkód / źródeł cieczy, proponuję zapoznać się z zamieszczonymi przykładami (pliki .grid). Istnieje bowiem kilka ogólnych zasad, których należy się trzymać by otrzymać rozwiązanie stabilne problemu.

Przede wszystkim definiowany problem musi być "rozsądny" - losowe rozmieszczenie komórek brzegowych i wpływowych / wypływowych na siatce na pewno nie da dobrych wyników.

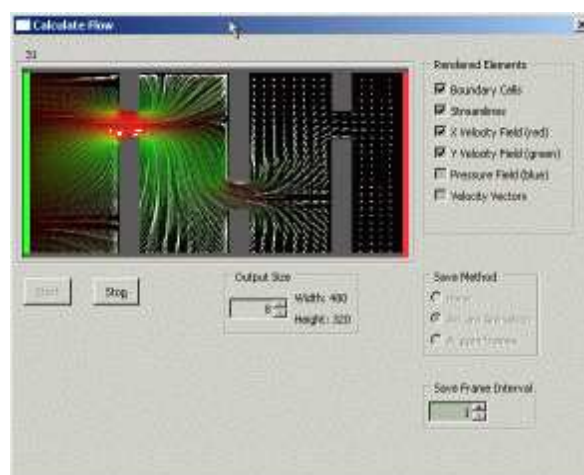
Podstawowe zasady, których należy się trzymać przy definiowaniu własnego problemu:

- Rozmiary siatki wybierać tak, by  $n_x \geq n_y$
- Oznaczyć komórki na brzegach siatki (najbardziej na lewo/prawo/do góry/na dół) jako:
  - Komórkę brzegową
  - Komórkę Wpływową
  - Komórkę Wypływową
- Komórki brzegowe wewnątrz obszaru symulacji stawiać podwójnie (pojedyncze ściany brzegowe przenoszą informacje)

W zakładce "Simulation Properties" możemy określić liczbę charakteryzującą rodzaj badanego przepływu (bezwymiarowa liczba Reynoldsa), a także krok czasowy symulacji. W przypadku wystąpienia niestabilności należy krok czasowy zmniejszać. Dodatkowo istnieje w programie również opcja ustawienia wartości prędkości początkowej ("odgadnięcie" pola prędkości) przez ręczne ustawienie prędkości poziomej i pionowej w zakładce "Simulation Properties". Ustawienie prędkości poziomej = 1.0 przydaje się w przypadku badania przepływów typu ciało w tunelu aero dynamicznym.

#### 4.1.2 Okno Obliczeń

Po stworzeniu siatki obliczeniowej używając przycisku "z wykrzyknikiem" lub przycisku "Calculate It" przechodzimy do okna obliczeń (rysunek 5), gdzie definiujemy sposób wizualizacji oraz zapisu plików wyników.



Rysunek 5: Okno Obliczeniowe.

Do wyboru użytkownika pozostaje sposób wizualizacji przepływu, gdzie można dowolnie łączyć wybrane sposoby wizualizacji (patrz sekcja z rezultatami).

Również wielkość obrazka dla plików wynikowych może być przez użytkownika zmieniana przez zmianę wartości w polu "Output Size". W przypadku rozdzielczości  $> 1$  procedura do wizualizacji interpoluje wartości z siatki obliczeniowej dla uzyskania rozmytego powiększenia.

Pole "Save Frame Interval" służy do ustalania co która klatka symulacji ma zostać zapisana w procesie tworzenia animacji/zapisywania klatek.

Uwaga: W przypadku wystąpienia błędu przy próbie zapisania animacji w formacie .avi należy zmniejszyć wielkość obrazka wynikowego ("Output Size").

## 5 Na koniec..

Program "Hydro -Dynamica" napisany został w języku C++. Przy odrobinie silnej woli kod programu może zostać przekompilowany pod systemem Linux. Przenośność kodu uzyskana została przez użycie biblioteki FOX<sup>8</sup> do stworzenia interfejsu użytkownika.

Program w wersji wykonywalnej rozpowszechniany będzie na zasadzie freeware i dostępny będzie na mojej stronie domowej<sup>9</sup>.

Na płycie CD którą oddaję znajduje się program w dwóch wersjach językowych, przykłady siatek obliczeniowych, niniejsza dokumentacja, animacje wygenerowane programem oraz kod źródłowy programu. Proszę o nie rozpowszechnianie kodu źródłowego dołączonego do programu.

Życzę owocnej pracy z programem.

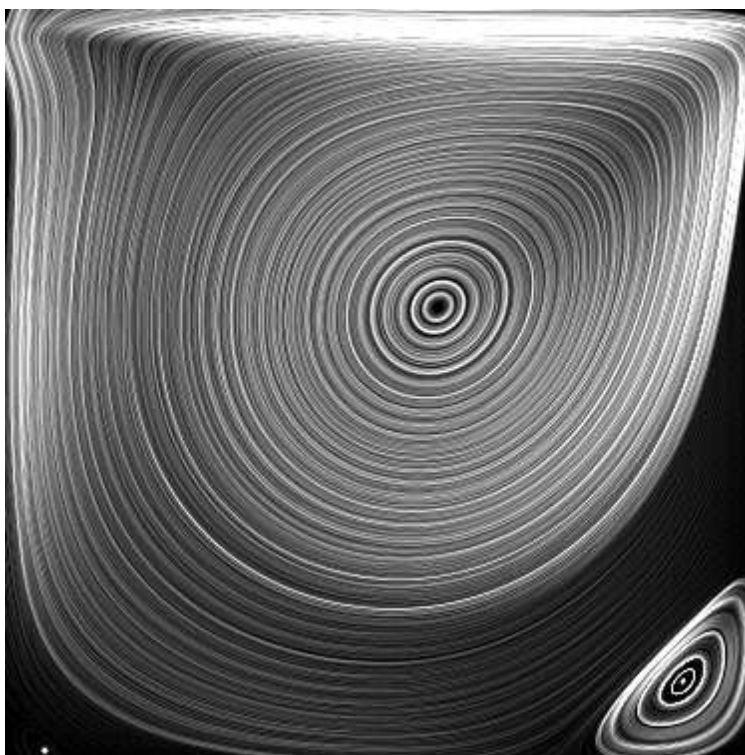
## 6 Rezultaty

W tej części dokumentacji prezentowane są wybrane problemy hydrodynamiki rozwiązane przy pomocy programu "Hydro -Dynamica".

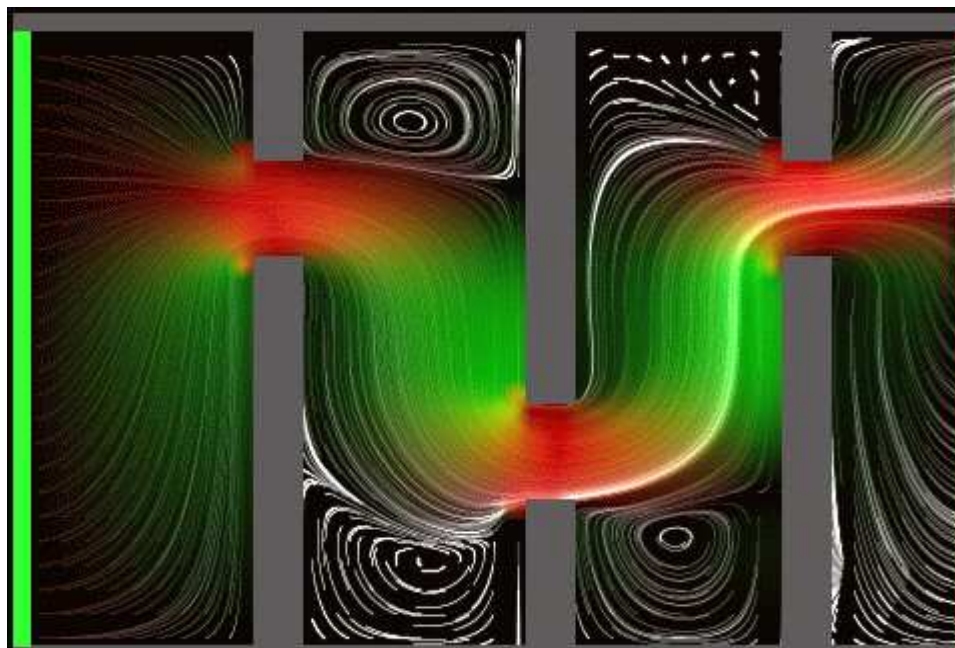
---

<sup>8</sup><http://www.fox-toolkit.org/>

<sup>9</sup><http://panoramix.ift.uni.wroc.pl/~maq>

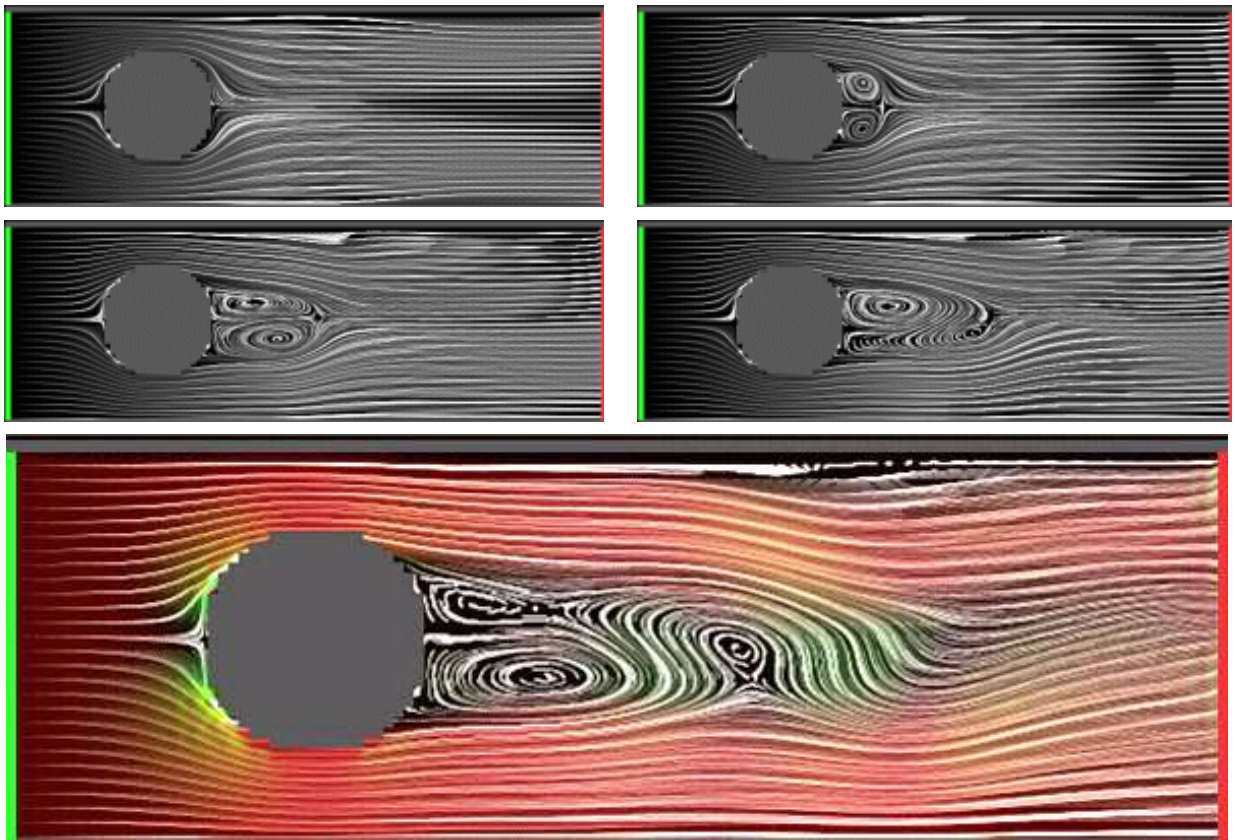


Rysunek 6: Klasyczny problem testowy - "jaskinia" z poruszającą się górną płytą. Charakterystyczne dwa wiry powstają dla liczby  $Re=400$ . Problem w oryginalnym nazewnictwie znany jako "Driven Cavity".



Rysunek 7: Przepływ cieczy o niskiej liczbie Reynoldsa  $= 10$  przez konfigurację trzech szczelin. Wizualizacja przez linie prądu oraz pola prędkości poziomej (kolor czerwony) i pionowej (kolor zielony).





Rysunek 8: Przepływ turbulentny cieczy dla  $Re=400$  w przypadku opływu przez "krążek". Ciecz za ciałem opływanym tworzy charakterystyczne wiry, znane w literaturze tematu jako "Vortex Karman Street".

## Literatura

- [1] John D. Anderson, Jr. 'Computational Fluid Dynamics: The Basics with Applications', McGraw-Hill Inc, 1995.
- [2] Ryszard Grybos, 'Podstawy mechaniki plynów' (Tom 1 i 2), PWN 1998.
- [3] David Potter 'Metody obliczeniowe fizyki', PWN 1982.
- [4] James D. Bozeman, Charles Dalton, 'Numerical Study of Viscous Flow in Cavity', Journal of Computational Physics, vol. 12, 1973.
- [5] J.C. Tannehill, D.A. Anderson, 'Computational Fluid Mechanics and Heat Transfer, Second Edition', Series in Computational and Physical Processes in Mechanics and Thermal Sciences .
- [6] C.A.J. Fletcher, 'Computational Techniques for Fluid Dynamics, Volume 2', Springer .
- [7] F. H. Harlow, John P. Shannon, 'The Splash of Liquid Drop', Journal of Applied Physics (vol. 38, n.10 Sept. 1967).
- [8] J. Welch, F. Harlow, J. Shannon, 'The MAC Method', Los Alamos Scientific Laboratory of the University of California (1965).
- [9] N.Foster, D.Metaxas, 'Realistic Animation of Liquids', Center for Human Modeling and Simulation.