

# Modelowanie Komputerowe

## Projekt - Elmer

Gabriela Białoskórska, Mikołaj Knysak, Ignacy Tekieli  
Fizyka Techniczna

Styczeń 2020

### Spis treści

<b>1</b>	<b>Wprowadzenie</b>	<b>2</b>
1.1	Cel pracy . . . . .	2
1.2	Wstęp teoretyczny . . . . .	2
1.2.1	Elmer . . . . .	2
1.2.2	Vortex Shedding . . . . .	3
<b>2</b>	<b>Materiały i Metody</b>	<b>5</b>
2.1	Wiadomości wstępne . . . . .	5
2.2	Tworzenie projektu . . . . .	6
2.2.1	Dodatkowe działania na symulacji . . . . .	8
<b>3</b>	<b>Wyniki</b>	<b>9</b>
3.1	Graficzna symulacja wyników . . . . .	11
<b>4</b>	<b>Podsumowanie</b>	<b>13</b>
<b>5</b>	<b>Bibliografia</b>	<b>13</b>

# 1 Wprowadzenie

## 1.1 Cel pracy

Celem niniejszej pracy było opracowanie zagadnienia *Vortex Shedding*, czyli modelowanie zjawiska generacji ścieżki wirowej von Karmana. Za obiekt symulacji obrana została rura z zawartą w niej okrągłą przeszkodą. Symulacja została wykonana przy użyciu oprogramowania Elmer.

## 1.2 Wstęp teoretyczny

W celu sukcesywnego wykonywania obliczeń inżynierskich zazwyczaj wykorzystuje się obliczenia polowe, przedstawiające przestrzenne rozkłady poszukiwanych wielkości. Umożliwia to poszukiwanie słabych punktów konstrukcji oraz ich ewentualną korektę, co zazwyczaj nie jest widoczne bezpośrednio.

W obecnych czasach najczęściej wykorzystujemy ku temu programy. Większość z nich należy do grupy programów komercyjnych, najczęściej płatnych, jednak istnieją również oprogramowania darmowe lub posiadające studenckie wersje demo z pewnymi ograniczeniami.

### 1.2.1 Elmer

Elmer jest systemem obliczeniowym stworzonym przez Finnish IT center for science. Instytucja ta udostępnia go zupełnie za darmo, jako formę eklamy swojego ośrodka. Rozwój Elmera to nieustający proces; jak każdy program zgodny z filozofią GNU, jego kod źródłowy jest szeroko dostępny. Wszelkie uwagi i spostrzeżenia są na bieżąco publikowane przez jego użytkowników na forum programu, a deweloperzy w szybki sposób starają się udzielać na nie odpowiedzi i nieść pomoc - niekiedy przygotowując nawet specjalne wersje oprogramowania. Elmer wspomaga badania naukowe w następujących dziedzinach:

- Modelowanie dynamiki płynów
- Obliczenia elektromagnetyczne
- Wymiana ciepła
- Akustyka
- Modelowanie odkształceń

Należy zaznaczyć, że jest to jeden z niewielu programów umożliwiających tego typu obliczenia.

W celu wykonania poprawnych obliczeń każdy program numeryczny wymaga przygotowania odpowiedniego pliku, bądź plików, w których zostanie opisane zagadnienie numeryczne w sposób zrozumiały dla danej aplikacji. Elmer wymaga przygotowania:

- Geometrii układu - konieczne jest stworzenie pliku z siatką
- Pliku z opisem problemu (.sif)

Elmer składa się z następujących programów:

- ElmerSolver – program rozwiązujący zagadnienie, zapisujący wyniki do pliku .ep (elmerpost)
- ElmerGrid – program do tworzenia siatki geometrii układu
- ElmerMesh2D - program do tworzenia siatki geometrii układu
- ElmerPost – program prezentujący wyniki obliczeń
- ElmerFront – graficzny interfejs programu
- ElmerGui – najnowsza wersja graficznego interfejsu użytkownika systemu Elmer
- matc – program matematyczny używany w systemie Elmer
- viewfactors – program obliczający współczynniki konfiguracji układu geometrycznego

Program Elmer może z początku wydawać się nieintuicyjny w obsłudze, jednak za darmo oferuje nam on pełny pakiet obliczeniowy, umożliwiający rozwiązywanie zaawansowanych problemów inżynierskich.

### 1.2.2 Vortex Shedding

W języku polskim nie istnieje odpowiednie tłumaczenie nazwy tego zjawiska. W dynamice płynów oznacza ono przepływ oscylacyjny, który ma miejsce gdy płyn, taki jak powietrze lub woda, przepływa obok urwiska z określoną prędkością, w zależności od wielkości i kształtu ciała. Wówczas wiry powstają z tyłu ciała i odrywają się okresowo z obu jego stron, tworząc **wir Von Karmana**.

Pojęcie wiru von Karmana funkcjonuje w literaturze ojczystej. Ma on następującą definicję: *Wir von Karmana polega na generacji wirów na przeszkodzie umieszczonej prostopadle do przepływającego medium (ciecz lub gaz)*. Jest ona zatem tożsama z definicją zjawiska opisywanego w języku angielskim (jak powyżej), który jest przedmiotem wykonywanej przez nas symulacji.

Zjawisko to od lat fascynowało uczonych, o czym mogą świadczyć na przykład szkice przedstawiające wiry formujące się na rzece narysowane przez Leonardo da Vinci. Cechą charakterystyczną opisywanych wirów jest to, że powstają one naprzemiennie raz z jednej raz z drugiej strony przeszkody, a częstotliwość

ich generacji jest wprost proporcjonalna do prędkości strugi. Inną bardzo istotną właściwością jest fakt, że częstotliwość wirów generowanych na przeszkodzie nie zależy od właściwości fizycznych płynu, a jedynie od jego prędkości. Zatem wynik pomiaru prędkości nie zależy od takich parametrów jak np. temperatura czy skład chemiczny medium.

### Modelowanie numeryczne zjawiska

Ruch lepkiego płynu newtonowskiego, nie posiadającego zdolności do akumulowania momentów skręcających można zapisać następująco:

$$\begin{aligned}\frac{dv_x}{dt} &= X - \frac{1}{\rho} \frac{\partial p}{\partial x} + \nu \left( \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} + \frac{\partial^2 v_z}{\partial z^2} \right) + \frac{\nu}{3} \frac{\partial}{\partial x} \left( \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right) \\ \frac{dv_y}{dt} &= Y - \frac{1}{\rho} \frac{\partial p}{\partial y} + \nu \left( \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} + \frac{\partial^2 v_z}{\partial z^2} \right) + \frac{\nu}{3} \frac{\partial}{\partial y} \left( \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right) \\ \frac{dv_z}{dt} &= Z - \frac{1}{\rho} \frac{\partial p}{\partial z} + \nu \left( \frac{\partial^2 v_x}{\partial x^2} + \frac{\partial^2 v_y}{\partial y^2} + \frac{\partial^2 v_z}{\partial z^2} \right) + \frac{\nu}{3} \frac{\partial}{\partial z} \left( \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} \right)\end{aligned}$$

gdzie:

$v_x, v_y, v_z$  – składowe jednostkowe wektora prędkości,  
 $X, Y, Z$  – składowe jednostkowe wektora siły masowej,  
 $\rho$  – gęstość płynu,  
 $\nu$  – kinematyczny współczynnik lepkości

Przedstawiony układ równań określany jest mianem **równań Naviera-Stokesa** i w wersji ogólnej wektorowej możemy go zapisać jako:

$$\frac{d\mathbf{v}}{dt} = \mathbf{F} - \frac{1}{\rho} \mathbf{grad}(p) + \nu \nabla^2 \mathbf{v} + \frac{\nu}{3} \mathbf{grad}(\mathbf{div}(\mathbf{v}))$$

gdzie:  $\mathbf{F}$  – wektor siły masowej

Uwzględniając równanie ciągłości:

$$\frac{dp}{dt} + \rho \mathbf{div}(\mathbf{v}) = 0$$

oraz fakt, że dla płynu nieściśliwego  $\rho = const.$ , otrzymujemy:

$$\mathbf{div}(\mathbf{v}) = \frac{\partial v_x}{\partial x} + \frac{\partial v_y}{\partial y} + \frac{\partial v_z}{\partial z} = 0$$

Po podstawieniu powyższej zależności do równania Naviera-Stokesa otrzymuje się **ogólne równanie przepływu dla nieściśliwego płynu newtonowskiego**:

$$\frac{d\mathbf{v}}{dt} = F - \frac{1}{\rho} \text{grad}(p) + \nu \nabla^2 \mathbf{v}$$

Równanie to wraz z równaniem ciągłości oraz odpowiednimi warunkami brzegowymi, daje zamknięty układ równań. Nie oznacza to jednak, że taki układ równań jest łatwy w rozwiązaniu. Clay Mathematics Institute zaliczył problem analitycznego rozwiązania równań Naviera-Stokesa do siedmiu najważniejszych nierozwiązanych problemów matematycznych i wyznaczył nagrodę w wysokości 1 mln USD za podanie kompletnego rozwiązania lub kontrprzykładu. Jak dotąd istnienie rozwiązania zostało udowodnione jedynie dla przypadku dwuwymiarowego

## 2 Materiały i Metody

### 2.1 Wiadomości wstępne

W celu wykonania symulacji opisywanych wirów, wykorzystaliśmy plik *ElmerTutorials.pdf* (udostępniony na zajęciach). Wybrane przez nas zagadnienie (Tutorial 8 - Vortex shedding – von Karman instability) posiadało gotową siatkę z obszarem symulacji, umieszczoną w archiwum.

Aby poprawnie wykonać zadanie, korzystaliśmy z informacji zawartych w powyżej wspomnianym tutorialu. Ich dokładny opis w języku polskim zawiera się w kolejnej sekcji (*Tworzenie projektu*). Początkowo usiłowaliśmy działać na systemie MacOS, jednak nie było to skuteczne, gdyż jak możemy przeczytać na oficjalnej stronie oprogramowania (<http://www.elmerfem.org/blog/binaries/>):

### Mac

There is no official Mac support. However, there are old recipes for homebrew that may help in the process.

<https://github.com/ElmerCSC/homebrew-elmerfem>

Wobec licznych, wynikających z tego problemów, zdecydowaliśmy się wykorzystać komputer z systemem Windows. Oszczędziło nam to wiele czasu i ułatwiło pracę.

## 2.2 Tworzenie projektu

Pracę nad projektem rozpoczęliśmy od wczytania podanego w tutorialu pliku z siatką `circle_in_channel.in2d`.

File

Open → `circle_in_channel.in2d`

Otrzymaliśmy wówczas siatkę składającą się z 749 węzłów oraz 1328 trójkątów. Zwiększyliśmy te wartości w następujący sposób:

Mesh

Configure

ngrid / Max H: 0.02 Mesh

Remesh

otrzymując 3464 węzłów i 6506 trójkątów.

Następnie, mając już gotową siatkę (Rys. 2), rozpoczęliśmy uzupełnianie wartości w menu Model. W Setup wybieraliśmy aspekty związane z symulacją, takie jak nazwy plików, odstępy czasu, stałe itp. Była ona przeprowadzana w dwuwymiarowych współrzędnych kartezjańskich. Metoda krokowa drugiego rzędu została wybrana za pomocą 200 kroków. Całkowity czas symulacji miał wynosić 8 sekund.

Model

Setup

Simulation Type = Transient

Steady state max. iter = 1

Time Stepping Method = bdf

BDF Order = 2

Time Step Intervals = 200

Time Step Sizes = \$ 8/200

W przypadku ustawień szczególnych dla solvera korzystaliśmy z wartości domyślnych. Złagodiliśmy natomiast tolerancję zbieżności w celu uzyskania szybszej symulacji.

Model

Equation

Name = Navier–Stokes

Apply to Bodies = 1

Navier–Stokes

Active = on

Edit Solver Settings

Nonlinear system

Convergence tol. = 1.0e−4

Linear System

Convergence tol. = 1.0e−6

Add

OK

Sekcja **Material** zawiera wszystkie parametry materiału. Wybraliśmy w niej proste parametry dla podstawowego przypadku testowego.

Model

```
Material
  Name = Ideal
  General
    Density = 1
  Navier Stokes
    Viscosity = 0.001
  Apply to Bodies = 1
  Add
  OK
```

System nie wymagał dodania żadnych sił. Pozostawiliśmy również domyślny warunek początkowy wynoszący zero. Każdy inny wymagałby jawnego ustawienia wartości.

Otrzymaliśmy trzy różne rodzaje granic: wlot, ściany antypoślizgowe i wylot. Wlot ma paraboliczny, w pełni rozwinięty profil laminarny z maksymalną prędkością 1,5 m/s. Dodatkowo dla wlotu przyjmuje się zerową składową prędkości pionowej. Okrąg oraz ściany dolna i górna są zabezpieczone antypoślizgową powłoką. Dla wylotu tylko składowa pionowa jest ustawiona na zero, ponieważ domyślna dyskretyzacja w słaby sposób nakłada warunek zerowego ciśnienia, jeśli normalna składowa prędkości nie jest zdefiniowana.

Model

```
BoundaryCondition
  Name = Inlet
  Navier-Stokes
    Velocity 1 = Variable Coordinate 2;
    Real MATC "4*1.5*tx*(0.41-tx)/0.41^2"
    Velocity 2 = 0.0
  Add
  New

  Name = Walls
  Navier-Stokes
    Velocity 1 = 0.0
    Velocity 2 = 0.0
  Add
  New

  Name = Outlet
  Navier-Stokes
    Velocity 2 = 0.0
  Add
  Ok
```

Warunki można także przypisać do granic w menu `Boundary condition` lub kliknięciem myszy. W tym przypadku zastosowaliśmy drugie podejście, ponieważ oszczędziło nam to wymogu znajomości indeksów każdej granicy.

Model

```
Set boundary properties
  Choose inlet → set boundary condition Inlet
  Choose both horizontal walls and circle →
  → set boundary condition Walls
  Choose outlet → set boundary condition Outlet
```

Do wykonania ElmerSolver potrzebowaliśmy plików siatki i pliku poleceń. Zasadniczo zdefiniowaliśmy wszystkie informacje potrzebne ElmerGUI do zapisania pliku poleceń. Po jego napisaniu mogliśmy także wizualnie obejrzeć plik poleceń.

Sif

```
Generate
Edit → look how your command file came out
```

Przed uruchomieniem solvera zapisaliśmy wszystkie pliki w katalogu. Projekt zawierał wszystko co było konieczne do ponownego uruchomienia symulacji.

File

```
Save Project
```

Po pomyślnym zapisaniu plików mogliśmy uruchomić solver

Run

```
Start solver
```

Widok zbieżności pojawia się automatycznie, pokazując względne zmiany każdej iteracji. Norma po pierwszym kroku powinna wynosić około 0,695, a po ostatnim 0,749. Posiadając wyniki, mogliśmy również uruchomić postprocesor.

Run

```
Start postprocessor
```

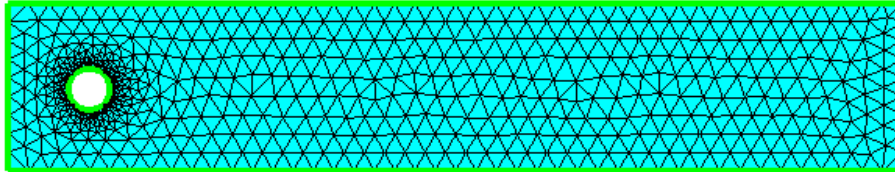
### 2.2.1 Dodatkowe działania na symulacji

Mając gotową symulację, utworzoną zgodnie z powyższymi wytycznymi, dokonaliśmy drobnej zmiany we własnym zakresie. Wynikało to z ciekawości w jaki sposób zachowa się symulacja w innym przypadku. Zmiana ta dotyczyła właściwości krawędzi - wpływ stał się odpływem, a odpływ wpływem. Sytuacja ta również została przez nas przedstawiona w sekcji "Wyniki" (Rys. 7).



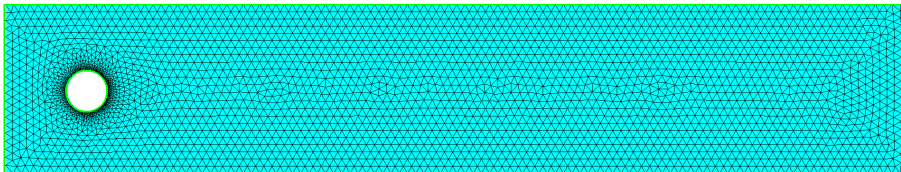
### 3 Wyniki

Wczytana przez nas na samym początku, domyślna siatka wyglądała następująco:



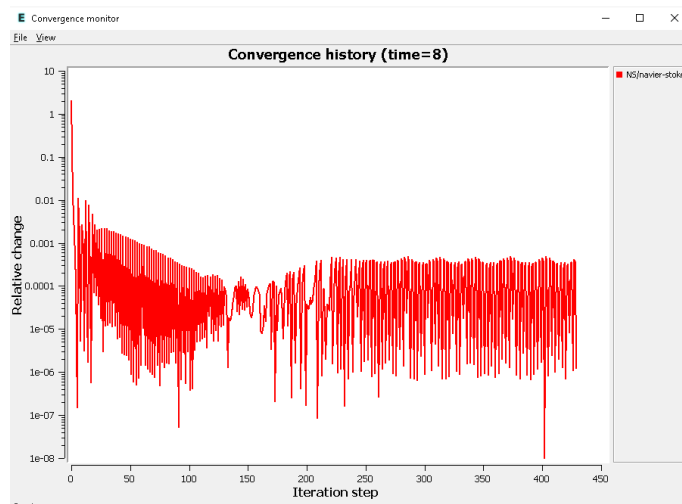
Rysunek 1: Domyślny wygląd "rzadkiej" siatki (749 węzłów oraz 1328 trójkątów)

Zgodnie z tutorialiem zmodyfikowaliśmy jej właściwości, zwiększając liczbę węzłów oraz trójkątów tak, aby była ona gęstsza.



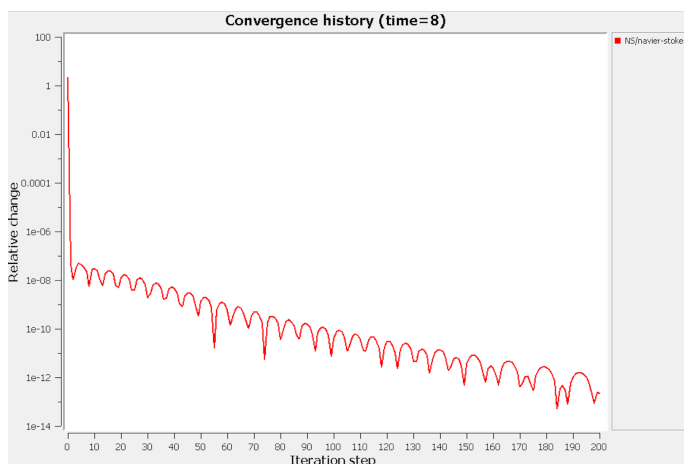
Rysunek 2: Nowy wygląd "gęstszej" siatki (3464 węzłów oraz 6506 trójkątów)

Następnie, mając już gotową siatkę, stworzyliśmy pierwszy solver, przedstawiony na poniższej ilustracji:



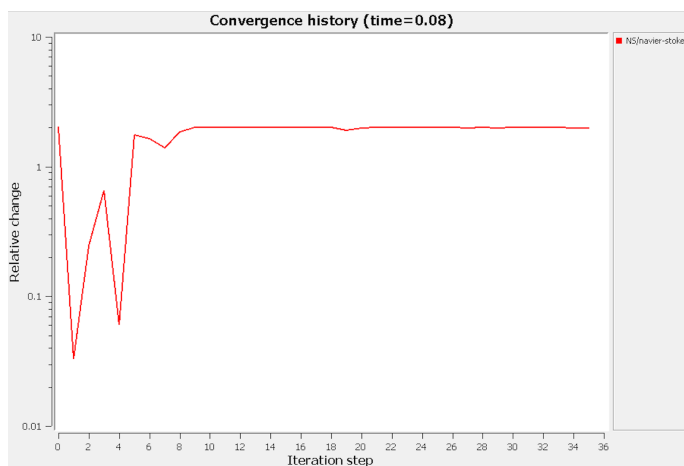
Rysunek 3: Pierwszy stworzony solver

W kolejnej części uzupełniliśmy symulację o własności materiału oraz równania, otrzymując:



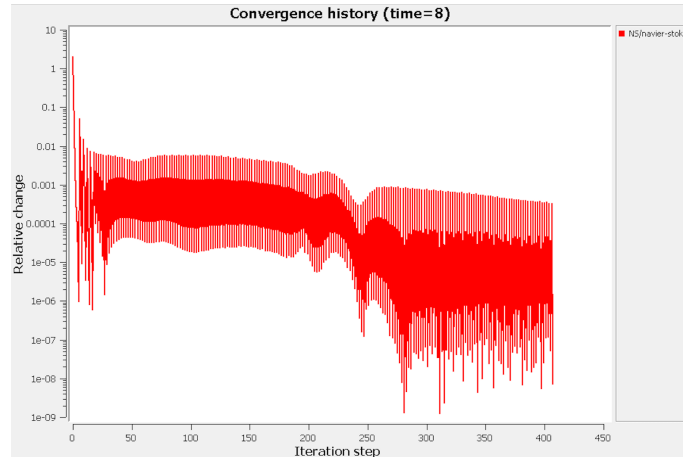
Rysunek 4: Wykres dla nowych właściwości

Następnie dodaliśmy warunek brzegowy dla wpływu:



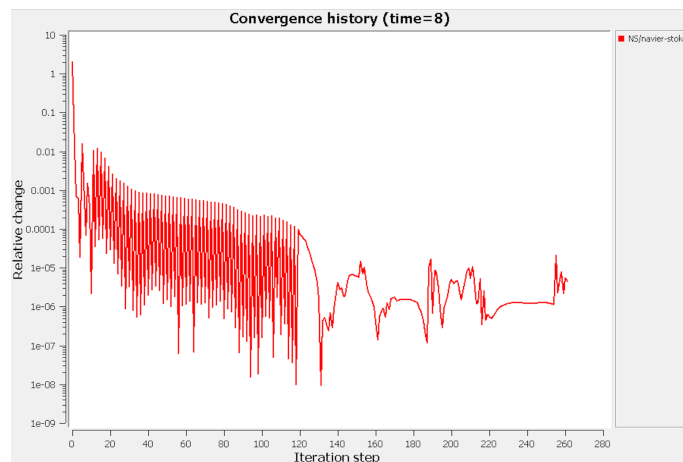
Rysunek 5: Wykres po dodaniu warunku brzegowego dla wpływu

Po dodaniu pozostałych warunków brzegowych wykres wyglądał następująco:



Rysunek 6: Wykres po dodaniu wszystkich warunków brzegowych

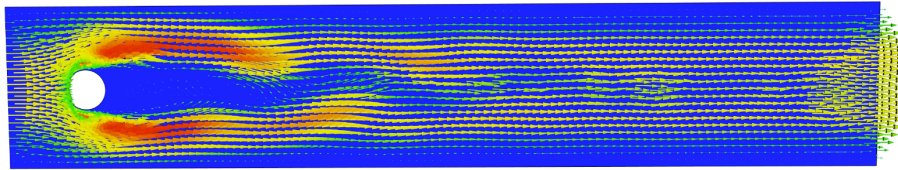
Zmiana wprowadzona do symulacji przez naszą grupę (rozdział 2.2.1 - zamiana wpływu na odpływ) w następujący sposób wpłynęła na zachowanie solwera:



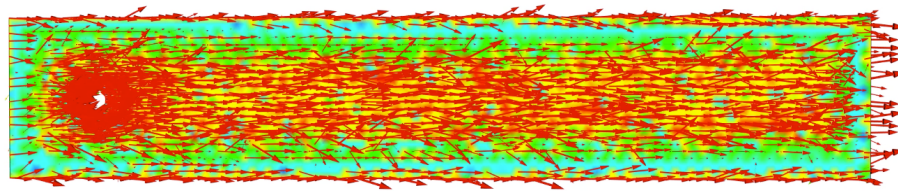
Rysunek 7: Wykres po wprowadzonej zmianie (zamiana wpływu na odpływ)

### 3.1 Graficzna symulacja wyników

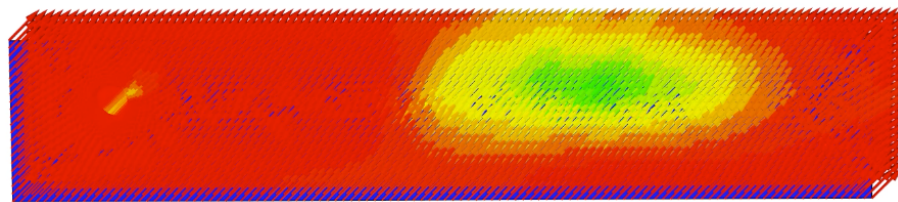
Powyższe wykresy mają rzecz jasna odwzorowanie w symulacji odbywającej się w danym przedziale czasu, tj. animacji. Poszczególne filmiki dołączamy do raportu, natomiast poniżej prezentujemy screeny z niektórych z nich, aby poglądowo ukazać w jaki sposób wyglądają one w Elmerze, a także zaznaczyć, że jedna z nich nie wyszła poprawnie (Rys. 9).



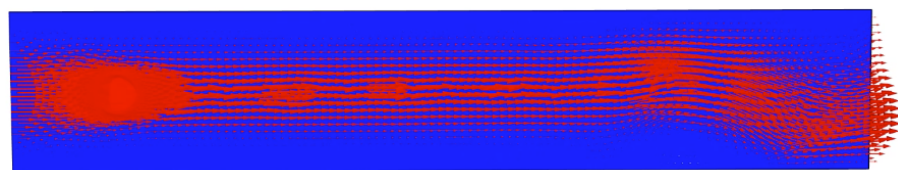
Rysunek 8: Animacja dla domyślnej siatki (zaraz po jej wczytaniu do programu)



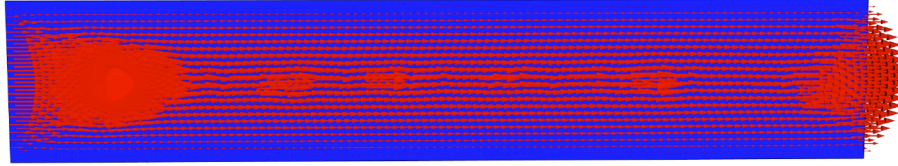
Rysunek 9: Animacja dla zmodyfikowanej siatki - błędny wynik



Rysunek 10: Animacja po wprowadzeniu właściwości materiału i równań



Rysunek 11: Animacja końcowa (wprowadzenie wszystkich warunków brzegowych)



Rysunek 12: Animacja po zmianie wpływu na odpływ

## 4 Podsumowanie

Symulacja dla większości zadanych warunków przebiegła pomyślnie, choć zdarzały się błędy, takie jak w przypadku animacji dla zmodyfikowanej siatki (Rys. 9). W ogólności jednak nie była ona skomplikowana do przeprowadzenia, zwłaszcza dysponując wspomnianym i opisanym w rozdziale drugim tutorialiem oraz gotową siatką na której podjęliśmy pracę.

Skorygowanie przedstawionych błędów wymaga od nas nieco więcej doświadczenia pracy z programem Elmer, na ten moment nie wiemy do końca gdzie je popełniliśmy. Mimo to staraliśmy się wprowadzić do projektu własną inicjatywę (rozdział 2.2.1) i zrobić nieco więcej, niż wymagała tego od nas instrukcja - niezależnie od wyników.

## 5 Bibliografia

- 1.) *"Obliczenia wymiany ciepła na drodze promieniowania z wykorzystaniem programu Elmer"* - Daniel Kucharski
- 2.) *"Modelowanie ścieżki wirowej von Karmana przy użyciu pakietu ANSYS FLUENT"* - Artur Kulińczak, Grzegorz Pankanin
- 3.) *"Elmer GUI Tutorials"* - CSC – IT Center for Science