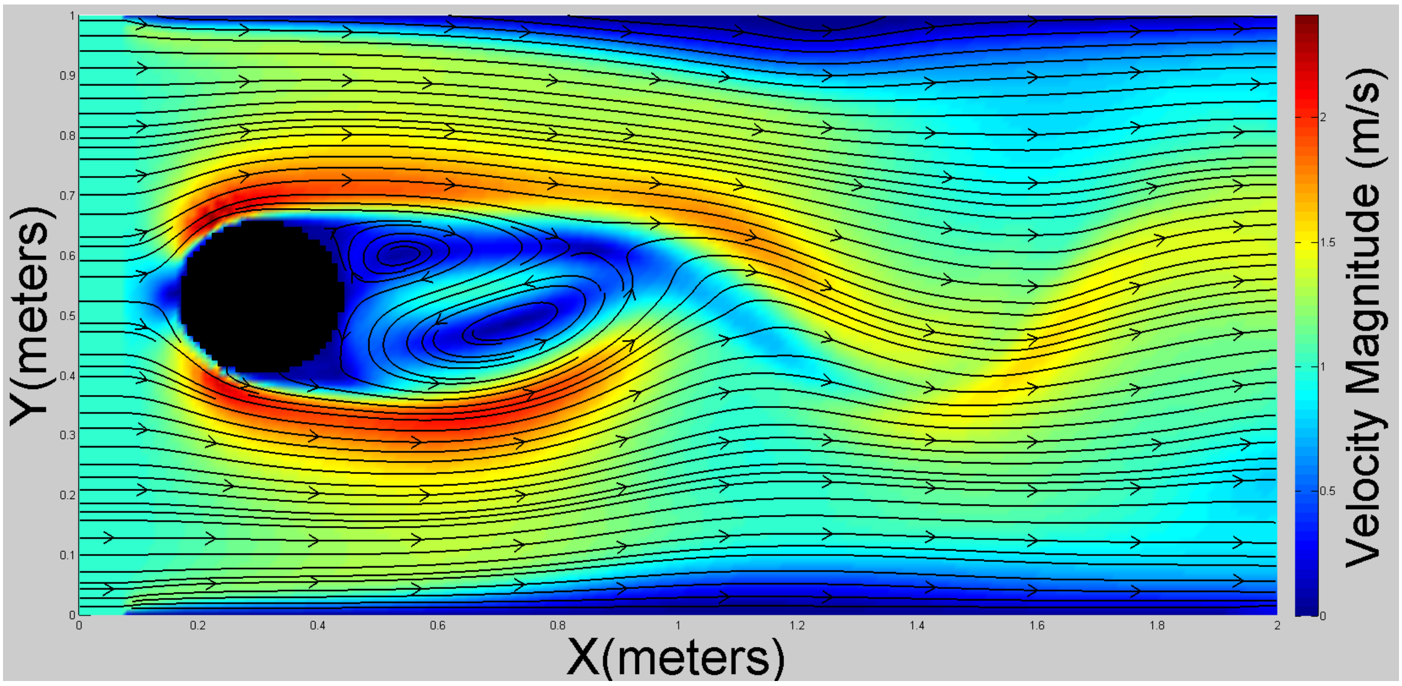
A matlab code for Numerical solution of Navier-stokes equations for two-dimensional incompressible flow (velocity-pressure formulation) along with ability for importing custom scenarios for the fluid flow.

Code Created by Jamie Johns 2018

$$\boxed{\begin{aligned}
&\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0\\
&\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{dP}{dx}+\vartheta\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)\\
&\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{dP}{dy}+\vartheta\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)
\end{aligned}}$$



Original source (github) of this document:

https://github.com/JamieMJohns/Navier-stokes-2D-numerical-solve-incompressible-flow-with-custom-scenarios-MATLAB-

Note: due to work and study commitments, this document is (for now) a very brief overview of the code and will eventually be updated with further detail about the algorithms applied. [date: 23/03/2018]

Quick description of the code:

- Code applies staggered-grid scheme for velocity, following the *Marker And Cell* Method (*MAC* Method). [sources of information on next pages].
- Code user can easily simulate custom scenarios of incompressible fluid flow by importing an image file (such as; jpeg, .png,.bmp etc..) along with modifying some parameters at the beginning of the code;
    - The code detects boundary conditions from the imported image (as well as what type of boundary conditions are to be used).
- The user can view resulting velocity fields in the form of still frames or an animated output, which can also be recorded as a video file (.avi file) of specified resolution, time-length and frame-rate.
- In addition, the code stores calculations for velocities (in segments of several ".mat" files) on local hard-drive as opposed to using computer memory (RAM), as calculations can be memory intensive;
    - During visualisation of calculated velocities, the code will automatically access the store files in sequence of when they were saved.
    - Provides the code user the opportunity to keep calculated velocities for later usage.

This readme provides;

- Sources of information for understanding the math/algorithm behind the code
- Demonstration of using the code (example scenarios)

Some other notes:

The code was originally created out of my own interest (rather than for the interest of a professional publication) and is subject to improvement, in particular; define a scheme (within the code) for improved handling and detection of when pressure instabilities which may occur (avoidance of divergent solutions for Pressure-Poisson equation).

For numerical solution of the pressure field (Poisson equation), a convergence criterion for the absolute difference between initial and current calculated pressure field of **0.001 is acceptable for most incompressible applications**** [in the code this parameter is named "error" (i.e – error=0.001)].

****source: https://www.flow3d.com/resources/cfd-101/numerical-issues/convergence-criteria/**

Outline of code algorithm

My code is inspired by MAC algorithm outlined in *https://www3.nd.edu/~gtryggva/CFD-Course2017/Lecture-5-2017.pdf*, which applied the algorithm to a standard "driven cavity" flow simulation, the algorithm which numerically solves 2D navier-stokes equations of incompressible flow in conservative form for a velocity-pressure formulation.

For my code, I have extended (and vectorised) the algorithm to handle custom (and more complex) fluid flow simulations for which the user can important the scenario as a image file and specify parameters at beginning of code such as;

->Magnitude and direction of velocity for regions of constant velocity.

->physical resolution of simulation (grid resolution)

->time resolution of simulation (delta time)

->fluid density

->dynamic viscosity

->and tolerance of error for solution to pressure poisson equation.

From importing an image (drawing of fluid flow scenario), the code detects conditions such as;

-> regions of free fluid flow (when a image pixel is (or close to) the white in colour).

->region of constant velocity (when a image pixel is close to red).

->regions of no fluid flow ("solid walls", when a image pixel is close to black).

->fluid flow out-lets (when an edge pixel of image is close to the colour green).

Several examples are provided on the pages below to give the user an idea about how to define a particular fluid flow simulation using an imported image as well defining parameters in the code.

The Naiver-Stokes equations for two-dimensional incompressible flow:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

$$\frac{\partial u}{\partial t}+u\frac{\partial u}{\partial x}+v\frac{\partial u}{\partial y}=-\frac{1}{\rho}\frac{dP}{dx}+\vartheta\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t}+u\frac{\partial v}{\partial x}+v\frac{\partial v}{\partial y}=-\frac{1}{\rho}\frac{dP}{dy}+\vartheta\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)$$

The above equations are solved in conservative form:

$$\frac{\partial u}{\partial x}+\frac{\partial v}{\partial y}=0$$

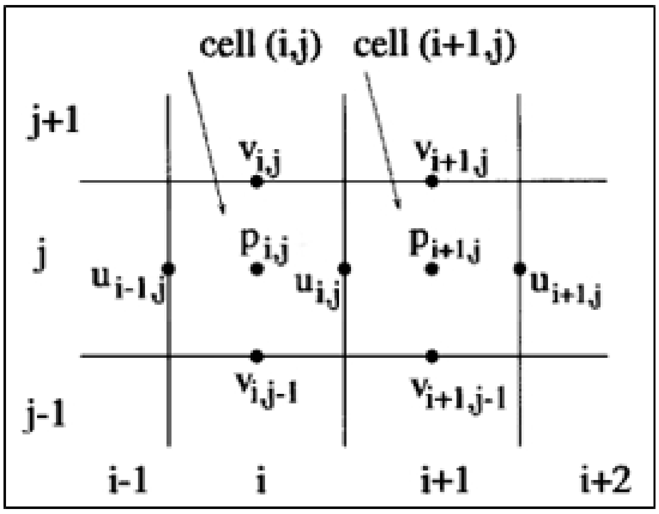
$$\frac{\partial u}{\partial t}+\frac{\partial u^2}{\partial x}+\frac{\partial vu}{\partial y}=-\frac{1}{\rho}\frac{dP}{dx}+\vartheta\left(\frac{\partial^2 u}{\partial x^2}+\frac{\partial^2 u}{\partial y^2}\right)$$

$$\frac{\partial v}{\partial t}+\frac{\partial uv}{\partial x}+\frac{\partial v^2}{\partial y}=-\frac{1}{\rho}\frac{dP}{dy}+\vartheta\left(\frac{\partial^2 v}{\partial x^2}+\frac{\partial^2 v}{\partial y^2}\right)$$

$(u,v)=(x,y)\mathit{component\ velocity}$

$t=\mathit{time}\ |\ \rho=\mathit{density}\ |\ \vartheta=\mathit{dynamic\ viscosity}\ |\ P=\mathit{pressure}$

Staggered grid is applied:



(image source: http://www.jmrt.com.br/en/simulation-unconstrained-solidification-a356-aluminum/articulo/S2238785413001129/)

List of files

Below is the complete list of files required to run the code. For quality control and to ensure that you are not running any corrupt code on your computer, make sure that these files are first downloaded from the original source that this document came from:

https://github.com/JamieMJohns/Navier-stokes-2D-numerical-solve-incompressible-flow-with-custom-scenarios-MATLAB-

Master file:

Main.m – main matlab file

Dependent files [these files are automatically used when running main.m]:

imagegrab.m – code that converts image files (i.e - .png,.jpg etc.) to data which is used in main.m

progressdisp.m – code that is used to display calculation progress in main.m

storevar.m – code that saves velocity data onto hardrive when running main.m (to avoid high RAM usage). The velocity data is stored as ".mat" files in directory "(matlab directory)\ temporary_NS_velocity" [this directory is automatically created when running .mat]/

openvar.m – code which retrieves (in correct sequence) files stored on to hardrive which are calculated velocity.

visualisation_still_frame.m– code for producing visualisation of velocity at single instant of time (frame); this file is used within main.m

visualisation_quick_animation.m– code for producing quick animated visualisation of velocity changing over time; this file is used within main.m

visualisation_record_video.m– code for producing animated visualisation of velocity changing over time along with the ability to record as video (.avi) file; this file is used within main.m

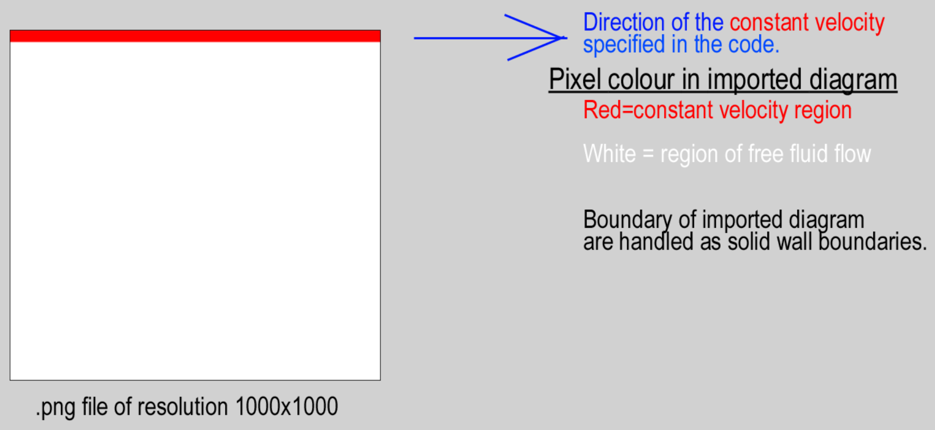
Additional files from github repository:

.png files – example scenariors of fluid flow [refer to example usage in next section of this document]

Example Usage of code

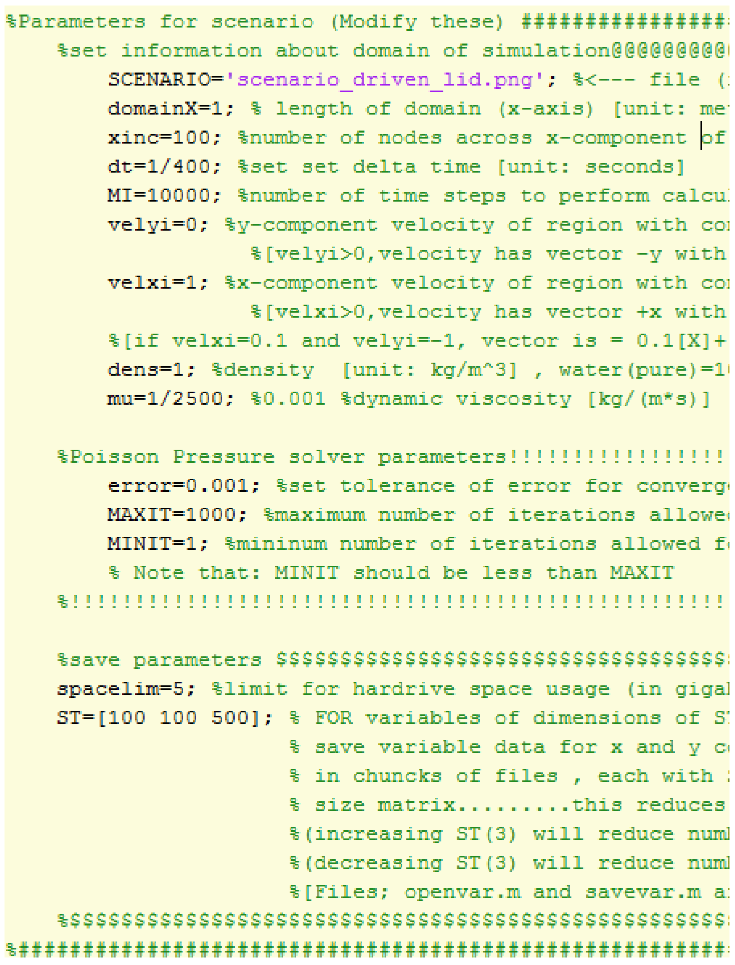
Example 1.

This fluid simulation is commonly known ‘Lid driven cavity flow’ and constist of a fluid trapped inside a box with the top wall (of the box) moving at a constant velocity which moves the fluid inside the box.



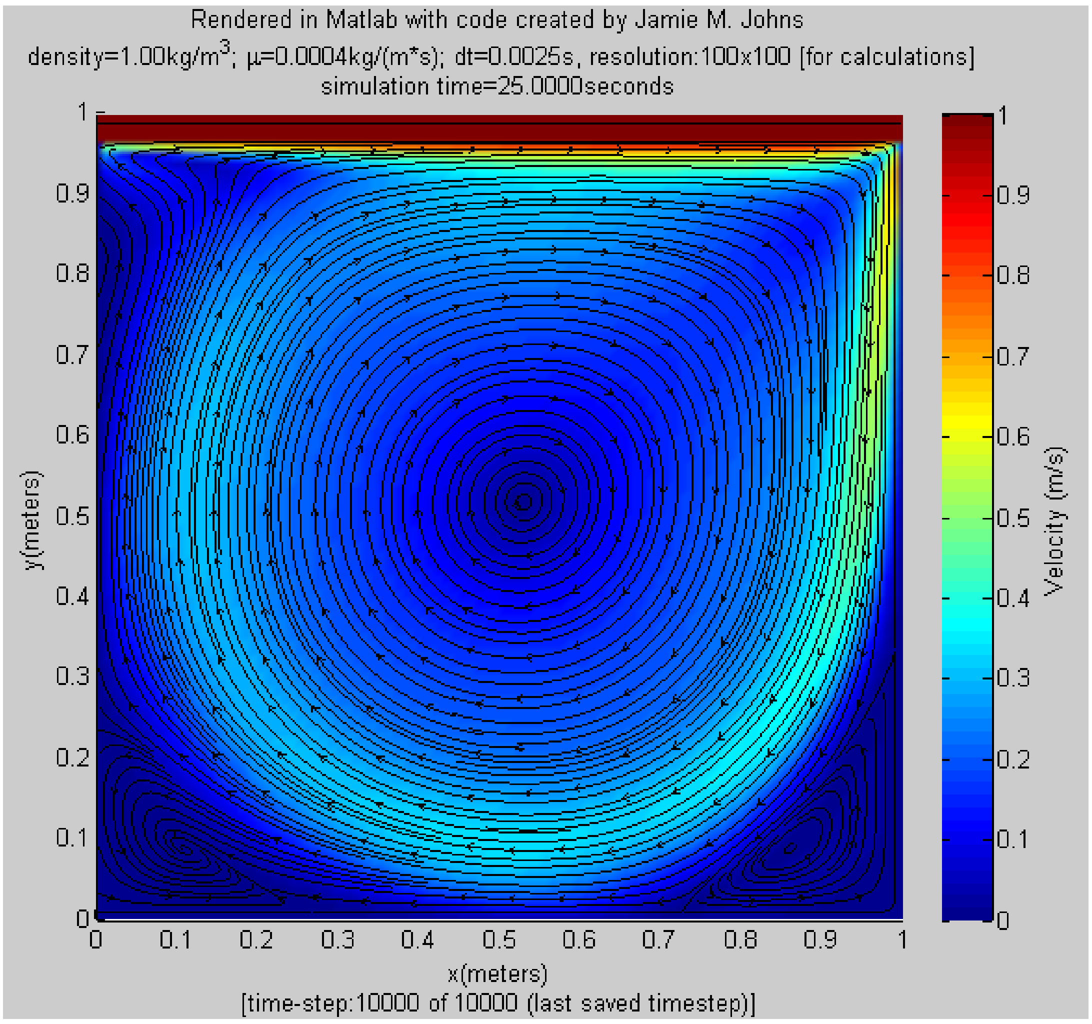
Direction of the constant velocity specified in the code.
Pixel colour in imported diagram
Red=constant velocity region
White = region of free fluid flow
Boundary of imported diagram are handled as solid wall boundaries.
.png file of resolution 1000x1000

Example settings



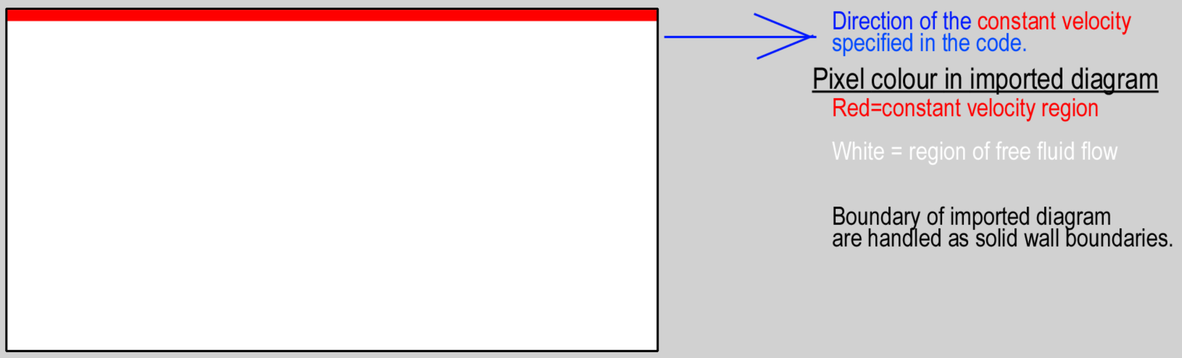
```
%Parameters for scenario (Modify these) ################
    %set information about domain of simulation@@@@@@@@
        SCENARIO='scenario_driven_lid.png'; %<--- file (
        domainX=1; % length of domain (x-axis) [unit: me
        xinc=100; %number of nodes across x-component of
        dt=1/400; %set set delta time [unit: seconds]
        MI=10000; %number of time steps to perform calcu
        velyi=0; %y-component velocity of region with co
                   %[velyi>0,velocity has vector -y with
        velxi=1; %x-component velocity of region with co
                   %[velxi>0,velocity has vector +x with
        %[if velxi=0.1 and velyi=-1, vector is = 0.1[X]+
        dens=1; %density  [unit: kg/m^3] , water(pure)=1
        mu=1/2500; %0.001 %dynamic viscosity [kg/(m*s)]

    %Poisson Pressure solver parameters!!!!!!!!!!!!!!!!
        error=0.001; %set tolerance of error for converg
        MAXIT=1000; %maximum number of iterations allowe
        MINIT=1; %mininum number of iterations allowed f
        % Note that: MINIT should be less than MAXIT
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage (in giga
    ST=[100 100 500]; % FOR variables of dimensions of S
                      % save variable data for x and y c
                      % in chuncks of files , each with
                      % size matrix.........this reduces
                      %(increasing ST(3) will reduce num
                      %(decreasing ST(3) will reduce num
                      %[Files; openvar.m and savevar.m a
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%#######################################################
```

Example output from settings on previous page



Example 2

Similar to example 1 but now the domain width is twice the length of the domain height.


Direction of the constant velocity specified in the code.
Pixel colour in imported diagram
Red=constant velocity region
White = region of free fluid flow
Boundary of imported diagram are handled as solid wall boundaries.

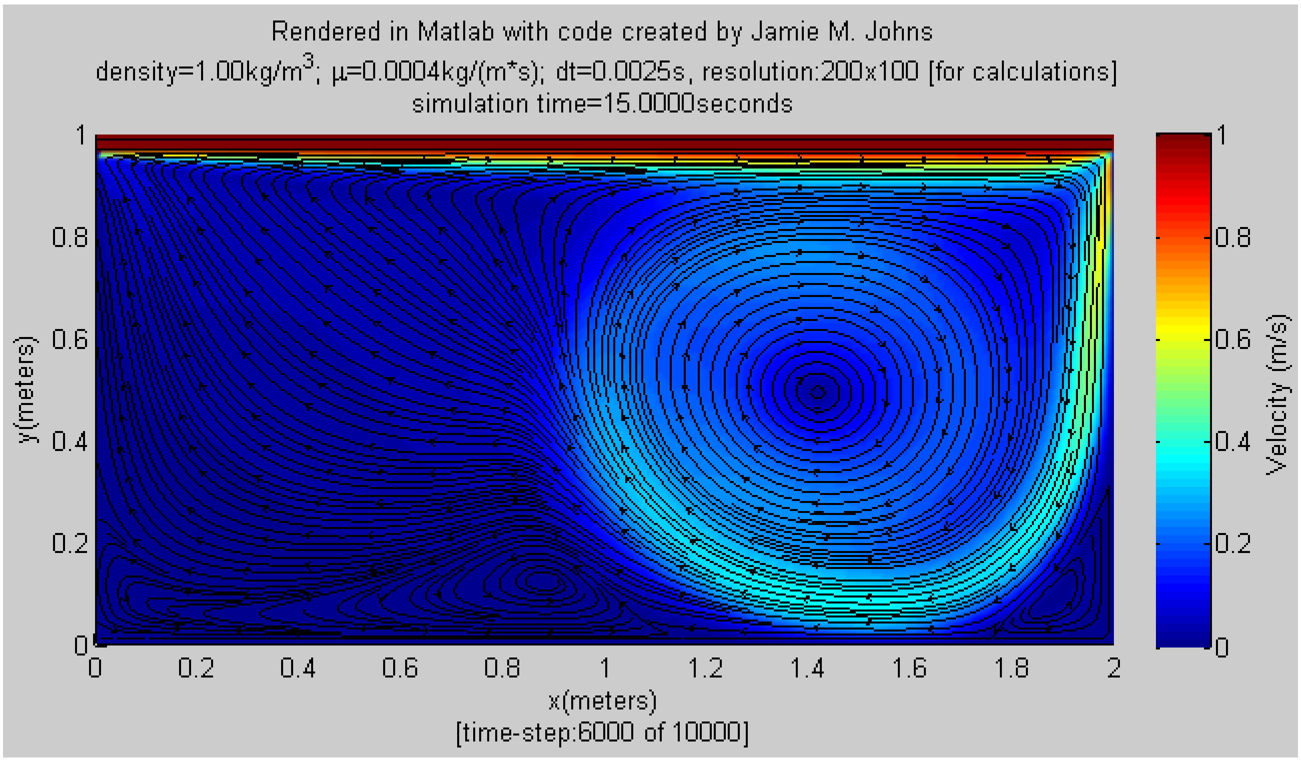
above: image that is imported into matlab

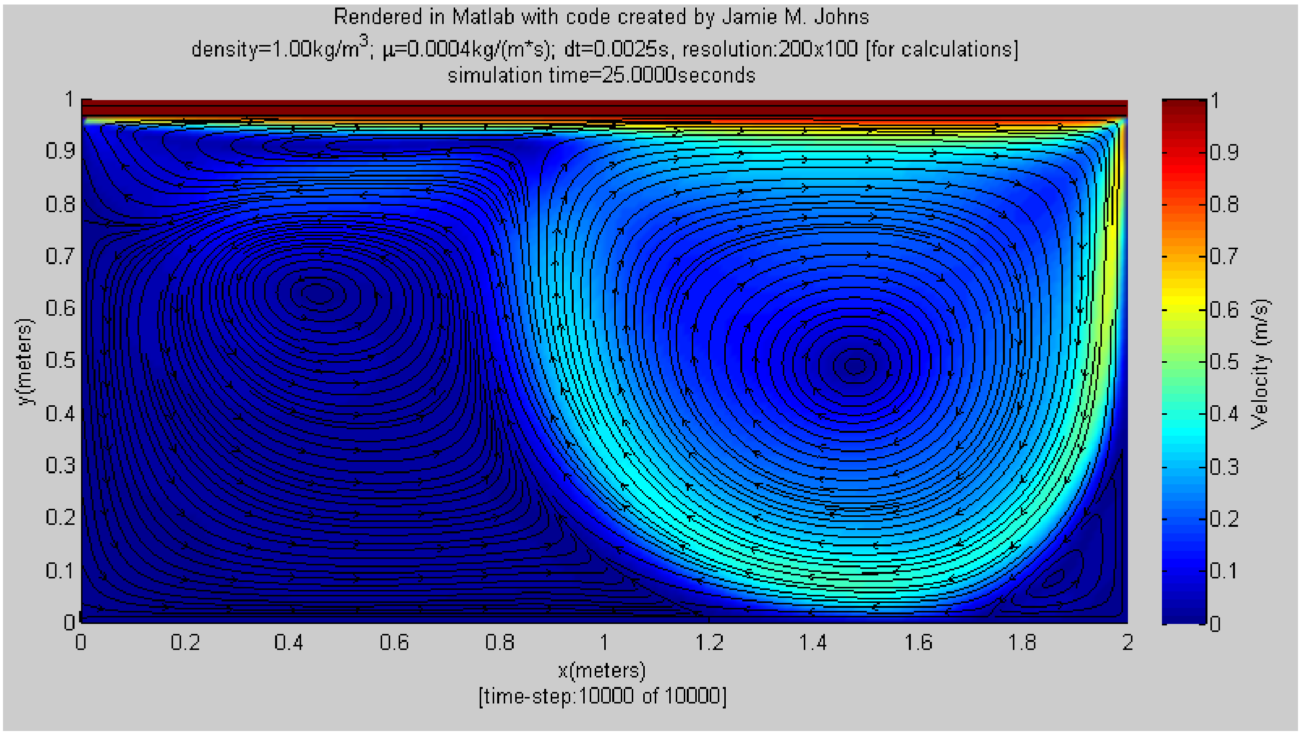
(.png file of resolution: 2000x1000)

Example settings

```
%Parameters for scenario (Modify these) ##############
    %set information about domain of simulation@@@@@@
        SCENARIO='scenario_driven_lid_long.png'; %<---
        domainX=2; % length of domain (x-axis) [unit:
        xinc=200; %number of nodes across x-component
        dt=1/400; %set set delta time [unit: seconds]
        MI=10000; %number of time steps to perform cal
        velyi=0; %y-component velocity of region with
                   %[velyi>0,velocity has vector -y wi
        velxi=1; %x-component velocity of region with
                   %[velxi>0,velocity has vector +x wi
        %[if velxi=0.1 and velyi=-1, vector is = 0.1[X
        dens=1; %density  [unit: kg/m^3] , water(pure)
        mu=1/2500; %0.001 %dynamic viscosity [kg/(m*s)

    %Poisson Pressure solver parameters!!!!!!!!!!!!!!!
        error=0.001; %set tolerance of error for conve
        MAXIT=1000; %maximum number of iterations allo
        MINIT=1; %mininum number of iterations allowed
        % Note that: MINIT should be less than MAXIT
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage (in gi
    ST=[100 100 500]; % FOR variables of dimensions of
                      % save variable data for x and y
                      % in chuncks of files , each wit
                      % size matrix.........this reduc
                      %(increasing ST(3) will reduce r
                      %(decreasing ST(3) will reduce r
                      %[Files; openvar.m and savevar.m
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%#####################################################
```

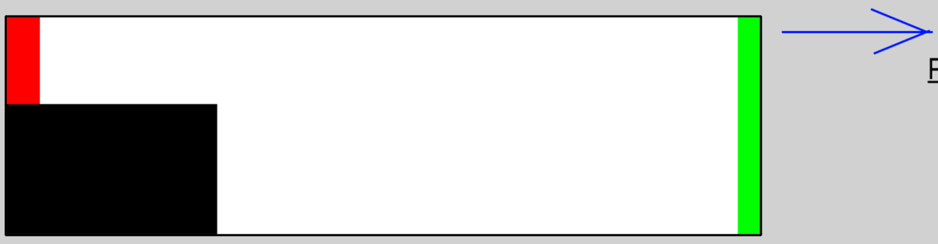
Example output from settings on previous page





Example 3

Simulation of fluid flow over a backward facing step.



Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

White = region of free fluid flow

Green = fluid outlet

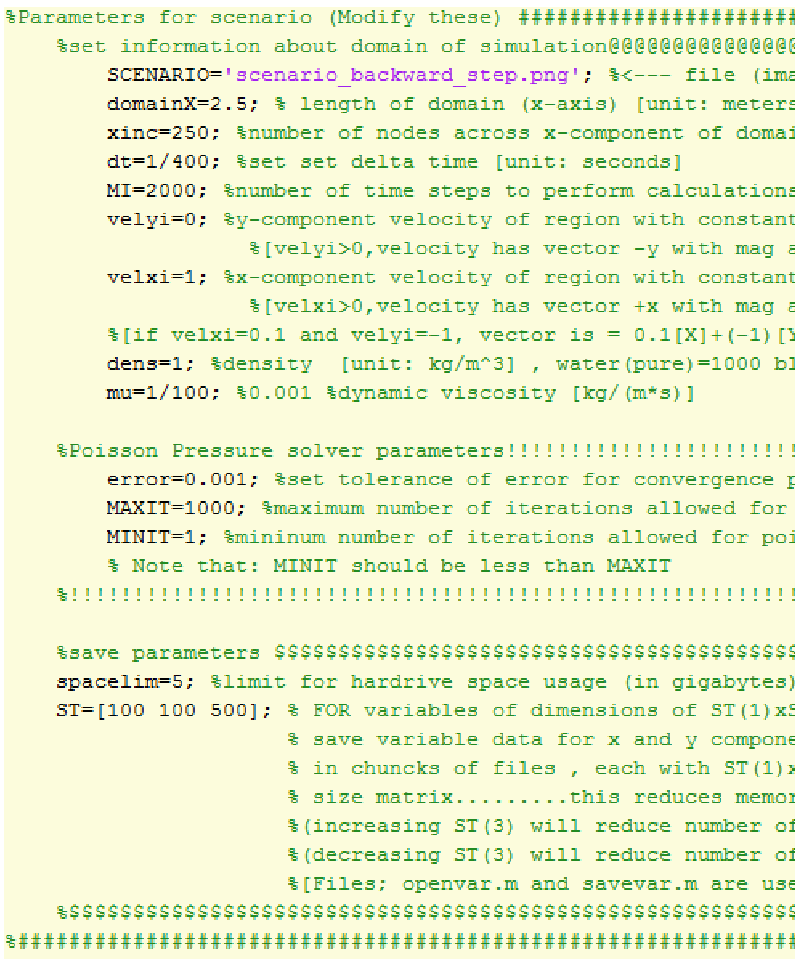
Black = region of no fluid flow ("solid wall")

Above: image that is imported into matlab

(.png file of resolution: 1500x500)

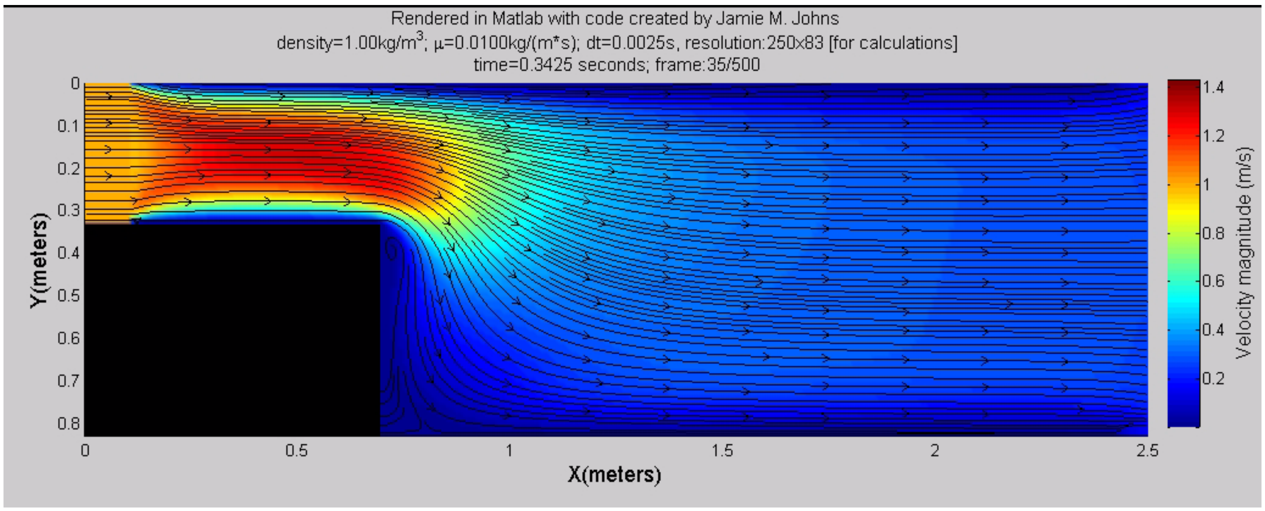
The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

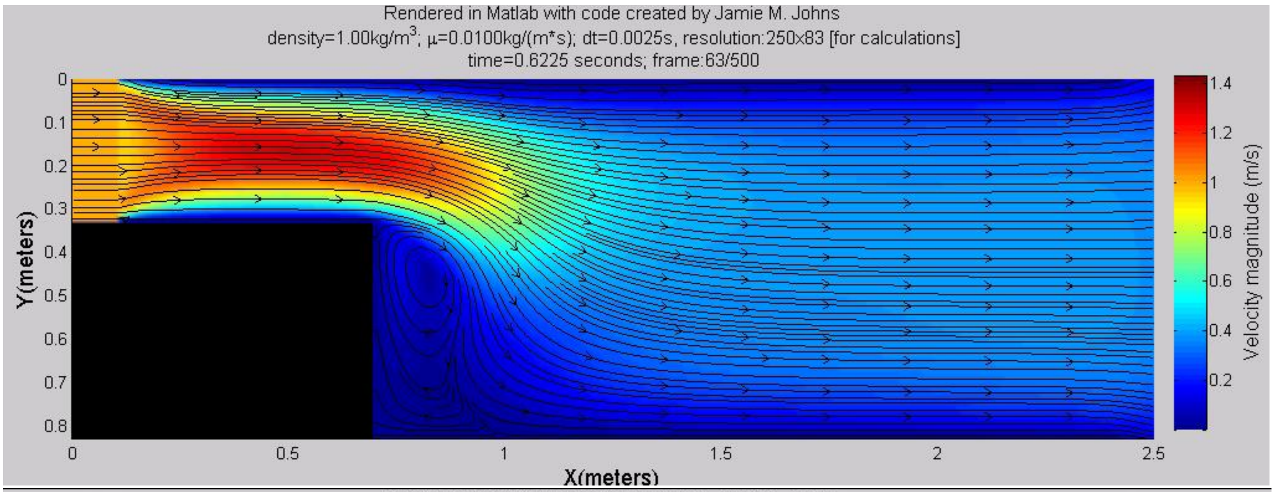
Example settings

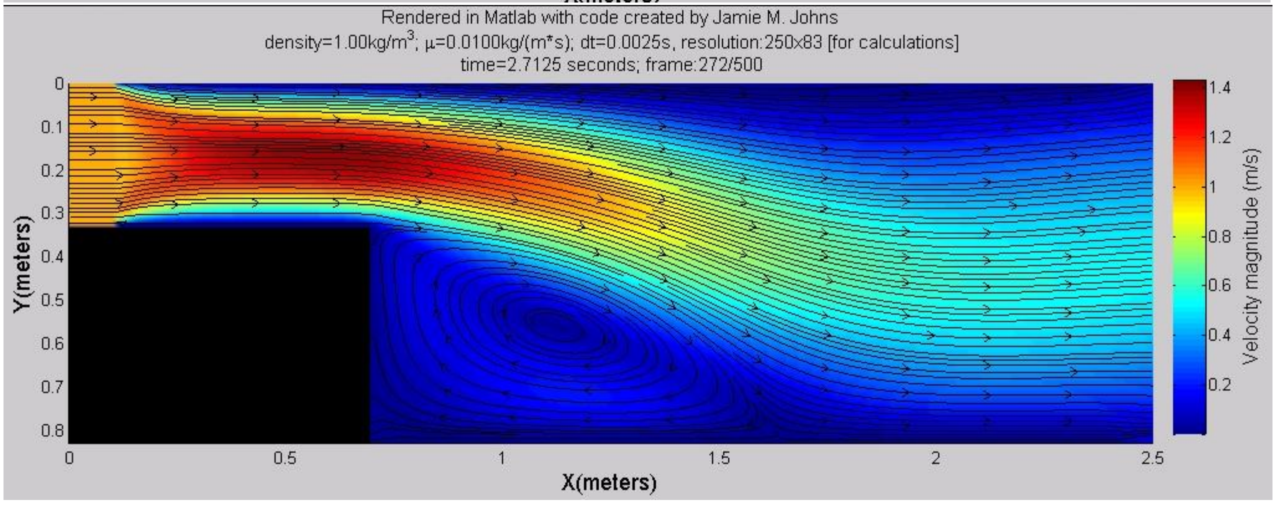


```
%Parameters for scenario (Modify these) ######################
    %set information about domain of simulation@@@@@@@@@@@@@@
        SCENARIO='scenario_backward_step.png'; %<--- file (ima
        domainX=2.5; % length of domain (x-axis) [unit: meters
        xinc=250; %number of nodes across x-component of domai
        dt=1/400; %set set delta time [unit: seconds]
        MI=2000; %number of time steps to perform calculations
        velyi=0; %y-component velocity of region with constant
                   %[velyi>0,velocity has vector -y with mag a
        velxi=1; %x-component velocity of region with constant
                   %[velxi>0,velocity has vector +x with mag a
        %[if velxi=0.1 and velyi=-1, vector is = 0.1[X]+(-1)[Y
        dens=1; %density  [unit: kg/m^3] , water(pure)=1000 bl
        mu=1/100; %0.001 %dynamic viscosity [kg/(m*s)]

    %Poisson Pressure solver parameters!!!!!!!!!!!!!!!!!!!!!!
        error=0.001; %set tolerance of error for convergence p
        MAXIT=1000; %maximum number of iterations allowed for
        MINIT=1; %mininum number of iterations allowed for poi
        % Note that: MINIT should be less than MAXIT
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage (in gigabytes)
    ST=[100 100 500]; % FOR variables of dimensions of ST(1)xS
                      % save variable data for x and y compone
                      % in chuncks of files , each with ST(1)x
                      % size matrix.........this reduces memor
                      %(increasing ST(3) will reduce number of
                      %(decreasing ST(3) will reduce number of
                      %[Files; openvar.m and savevar.m are use
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%##############################################################
```

Example output from settings on previous page

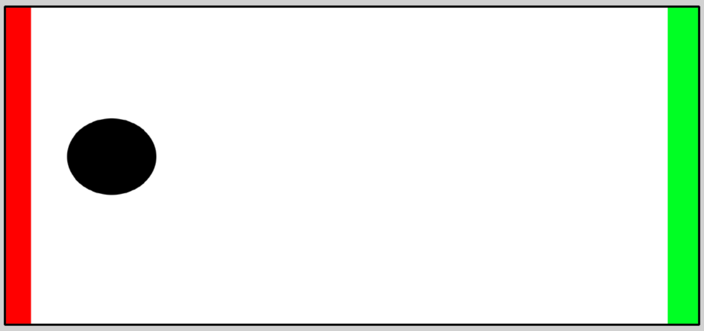






Example 4

Simulation of fluid flow around a two-dimensional sphere.



Above: image that is imported into matlab

(.png file of resolution: 2000x1000)

Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

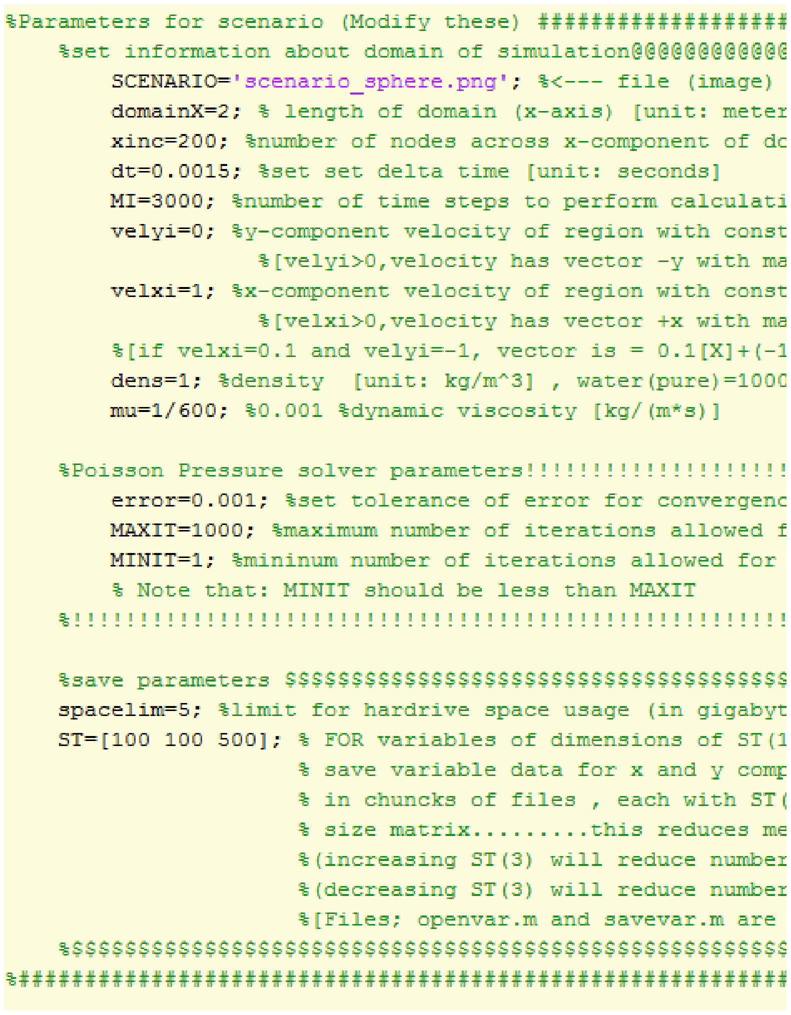
White = region of free fluid flow

Green = fluid outlet

Black = region of no fluid flow ("solid wall")

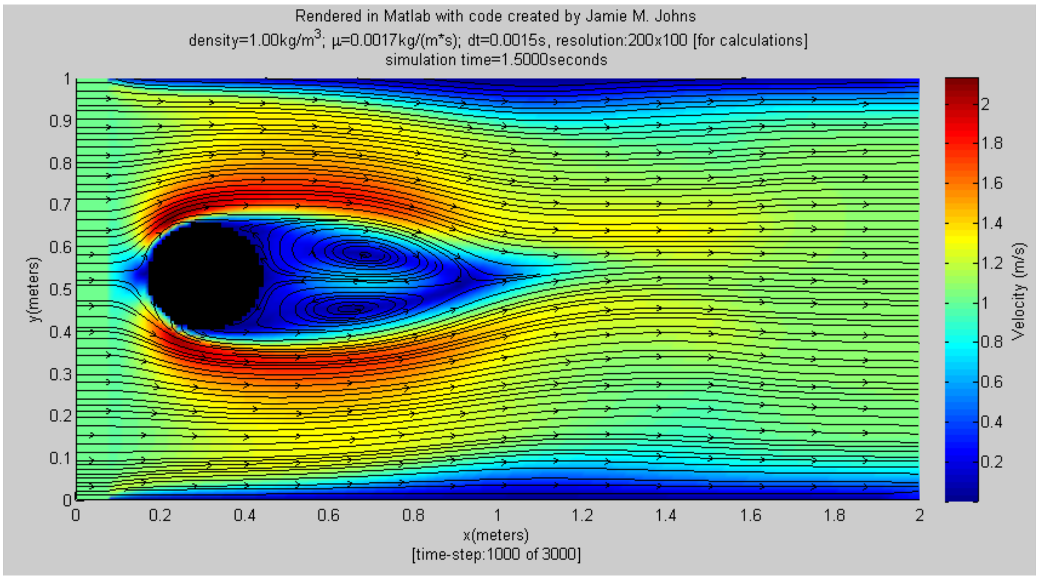
The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

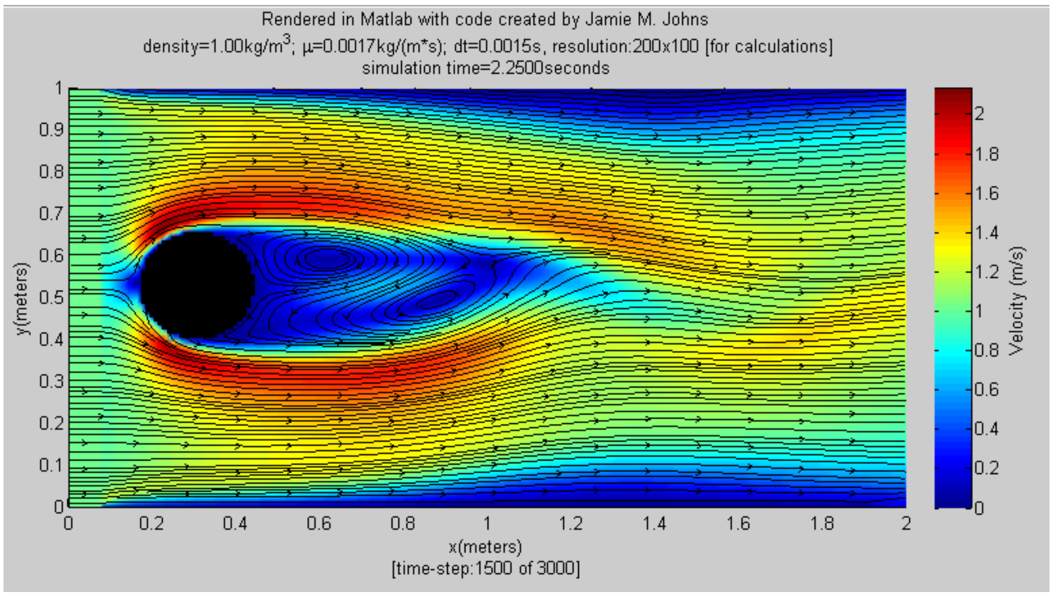
Example settings

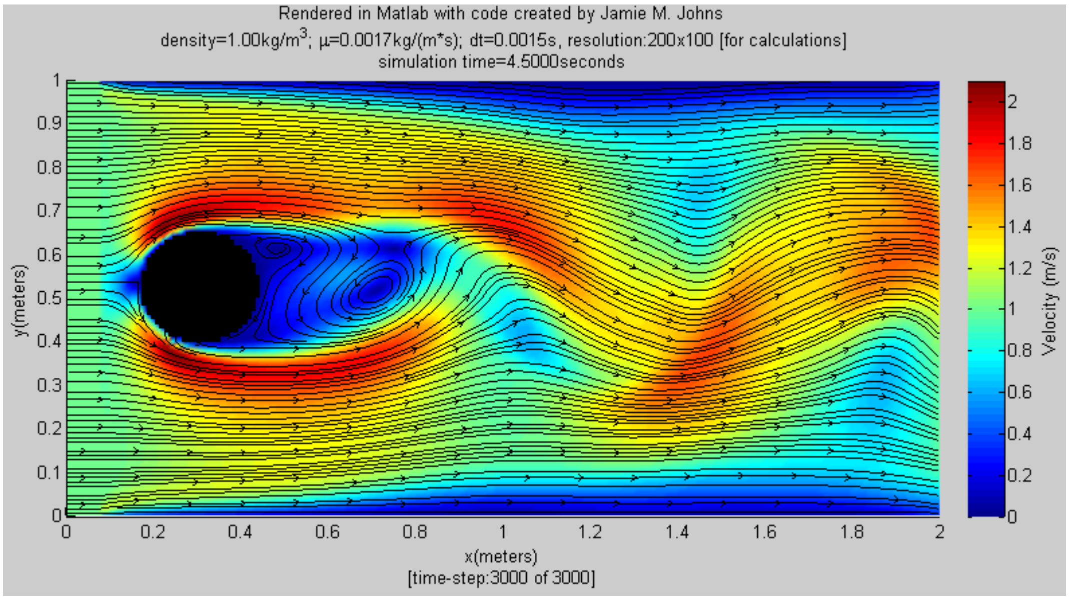


```
%Parameters for scenario (Modify these) ####################
    %set information about domain of simulation@@@@@@@@@@@@
        SCENARIO='scenario_sphere.png'; %<--- file (image)
        domainX=2; % length of domain (x-axis) [unit: meter
        xinc=200; %number of nodes across x-component of dc
        dt=0.0015; %set set delta time [unit: seconds]
        MI=3000; %number of time steps to perform calculati
        velyi=0; %y-component velocity of region with const
                   %[velyi>0,velocity has vector -y with ma
        velxi=1; %x-component velocity of region with const
                   %[velxi>0,velocity has vector +x with ma
        %[if velxi=0.1 and velyi=-1, vector is = 0.1[X]+(-1
        dens=1; %density  [unit: kg/m^3] , water(pure)=1000
        mu=1/600; %0.001 %dynamic viscosity [kg/(m*s)]

    %Poisson Pressure solver parameters!!!!!!!!!!!!!!!!!!!!
        error=0.001; %set tolerance of error for convergenc
        MAXIT=1000; %maximum number of iterations allowed f
        MINIT=1; %mininum number of iterations allowed for
        % Note that: MINIT should be less than MAXIT
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage (in gigabyt
    ST=[100 100 500]; % FOR variables of dimensions of ST(1
                      % save variable data for x and y comp
                      % in chuncks of files , each with ST(
                      % size matrix.........this reduces me
                      %(increasing ST(3) will reduce number
                      %(decreasing ST(3) will reduce number
                      %[Files; openvar.m and savevar.m are
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%##########################################################
```

Example output from settings on previous page

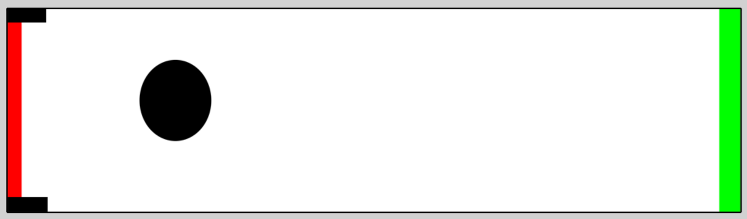






Example 5

Another simulation of fluid flow around a two-dimensional sphere.



Above: image that is imported into matlab

(.png file of resolution: 2000x500)

→ Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

White = region of free fluid flow

Green = fluid outlet

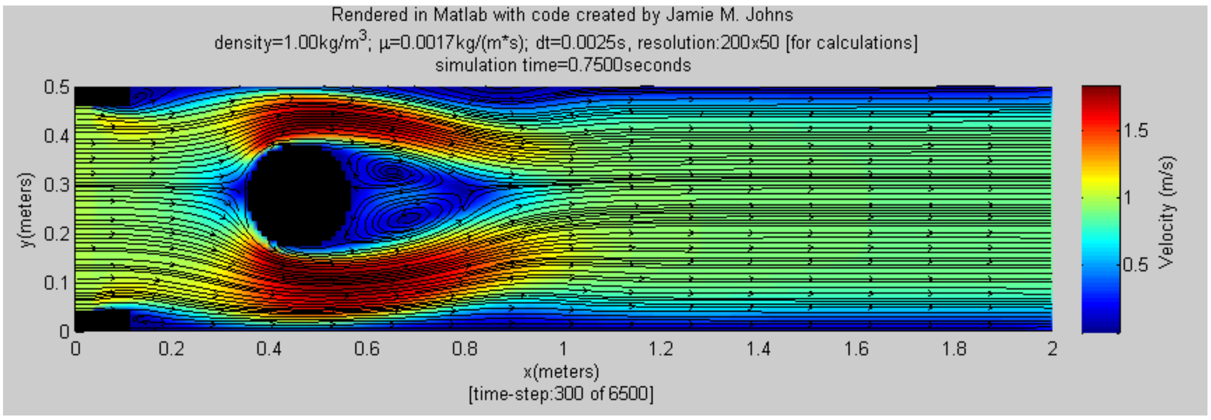
Black = region of no fluid flow ("solid wall")

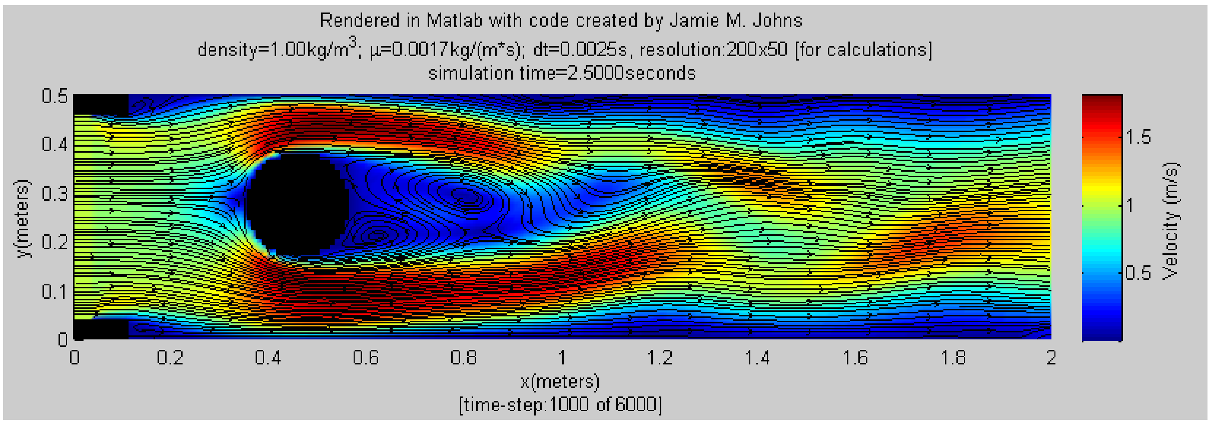
The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

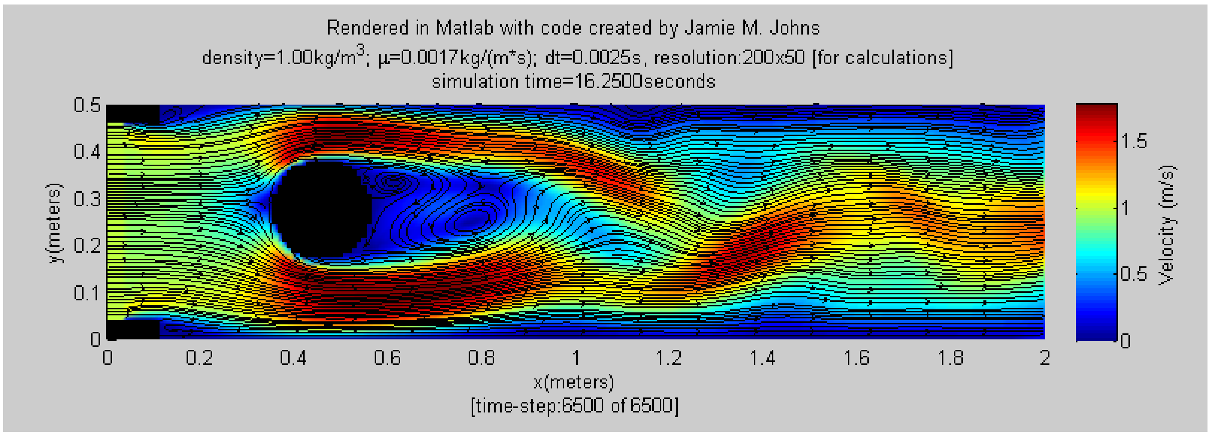
Example settings

```
%Parameters for scenario (Modify these) #########################
    %set information about domain of simulation@@@@@@@@@@@@@@@@@@
        SCENARIO='scenario_sphere2.png'; %<--- file (image) that
        domainX=2; % length of domain (x-axis) [unit: meters]
        xinc=200; %number of nodes across x-component of domain (
        dt=1/400; %set set delta time [unit: seconds]
        MI=6500; %number of time steps to perform calculations [t
        velyi=0; %y-component velocity of region with constant ve
                   %[velyi>0,velocity has vector -y with mag abs(
        velxi=1; %x-component velocity of region with constant ve
                   %[velxi>0,velocity has vector +x with mag abs(
        %[if velxi=0.1 and velyi=-1, vector is = 0.1[X]+(-1)[Y] a
        dens=1; %density  [unit: kg/m^3] , water(pure)=1000 blood
        mu=1/600; %0.001 %dynamic viscosity [kg/(m*s)]

    %Poisson Pressure solver parameters!!!!!!!!!!!!!!!!!!!!!!!!!
        error=0.001; %set tolerance of error for convergence pois
        MAXIT=1000; %maximum number of iterations allowed for poi
        MINIT=1; %mininum number of iterations allowed for poisso
        % Note that: MINIT should be less than MAXIT
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage (in gigabytes) fo
    ST=[100 100 500]; % FOR variables of dimensions of ST(1)xST(2
                      % save variable data for x and y component
                      % in chuncks of files , each with ST(1)xST(
                      % size matrix.........this reduces memory a
                      %(increasing ST(3) will reduce number of ex
                      %(decreasing ST(3) will reduce number of ex
                      %[Files; openvar.m and savevar.m are used]
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%################################################################
```

Example outputs from settings on previous page

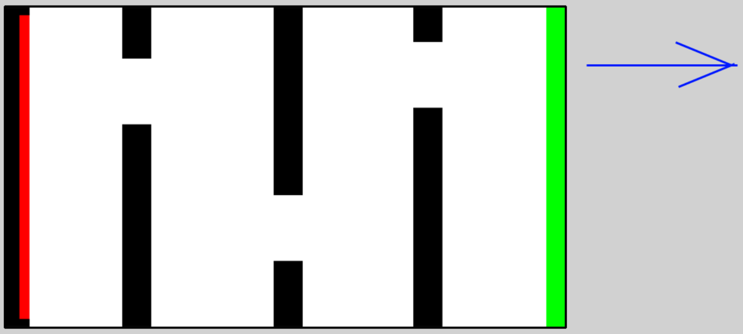






Example 6

Simulation of fluid flow through a very simple maze.



Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

White = region of free fluid flow

Green = fluid outlet

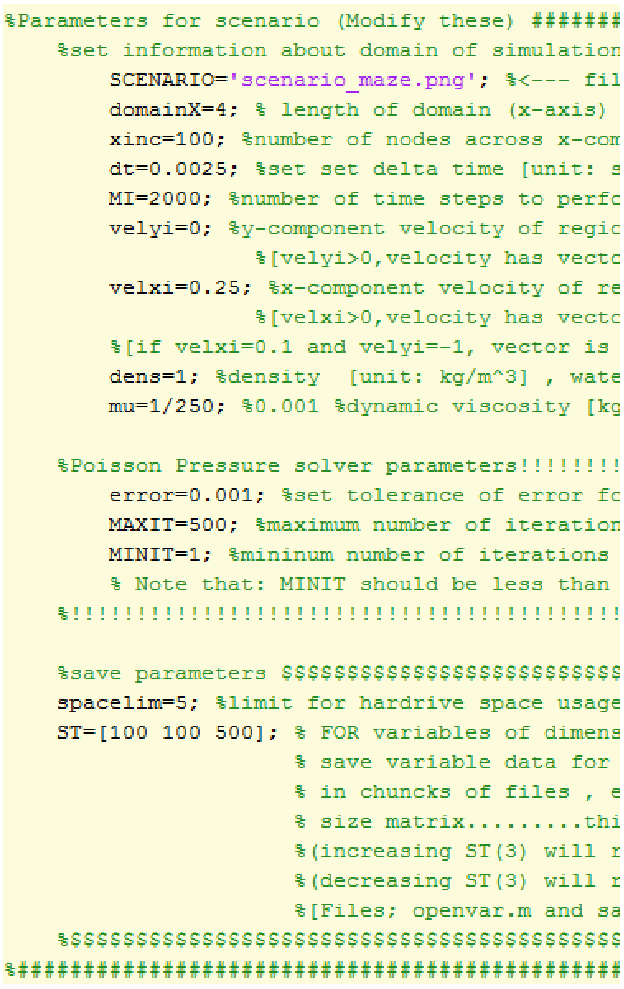
Black = region of no fluid flow ("solid wall")

Above: image that is imported into matlab

(.png file of resolution: 1000x500)

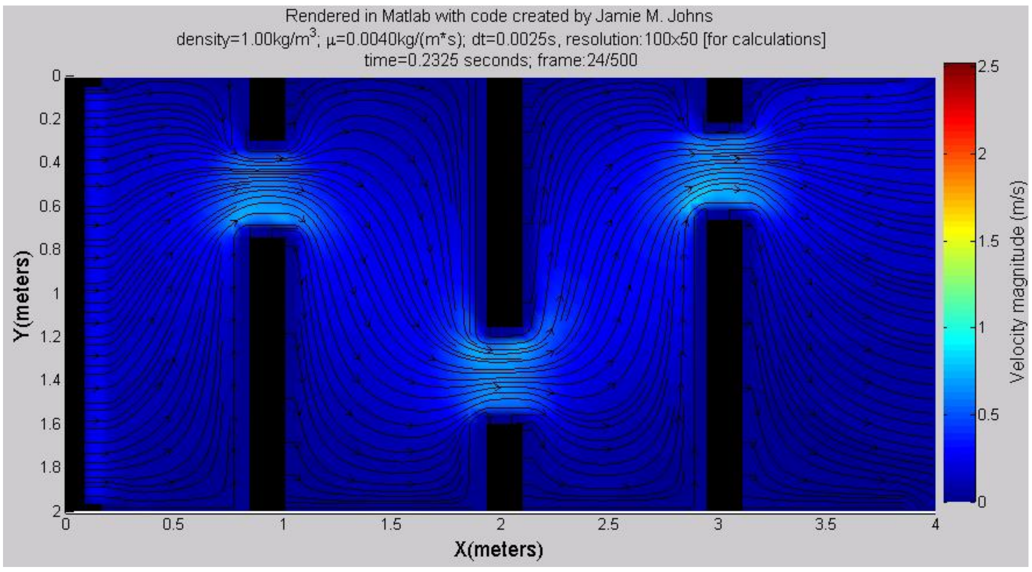
The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

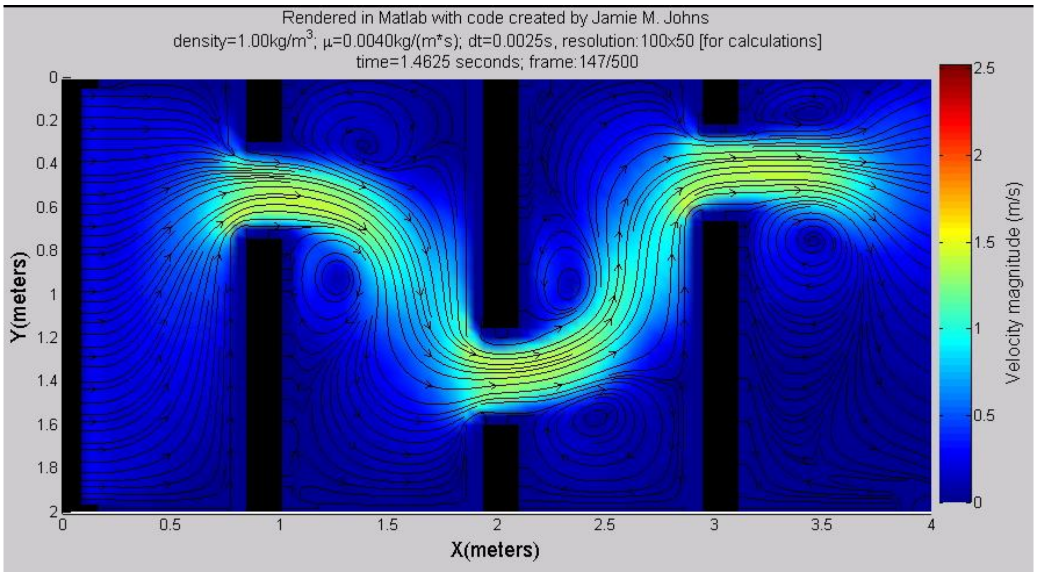
Example settings

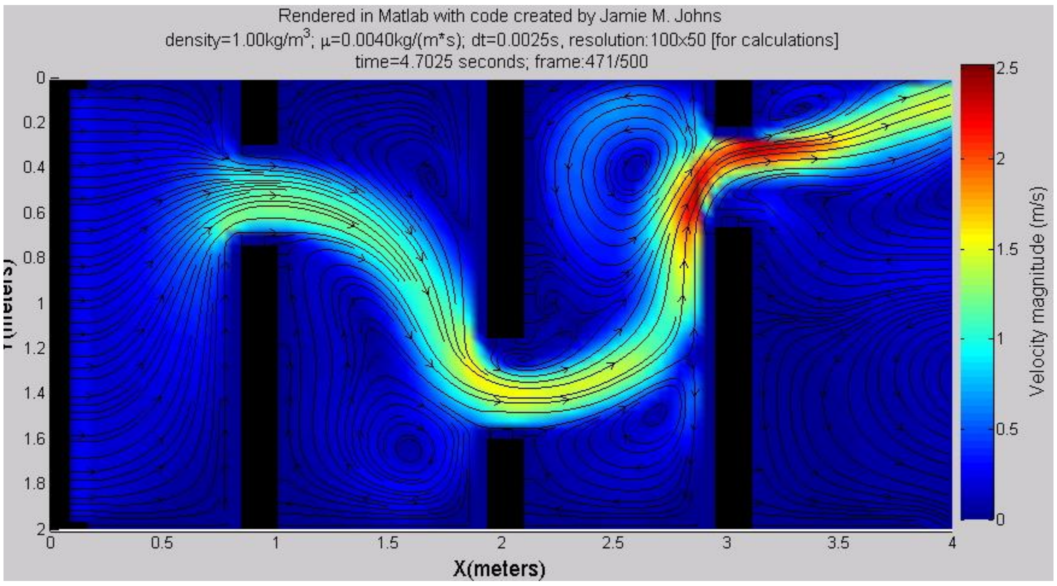


```
%Parameters for scenario (Modify these) #######
    %set information about domain of simulation
        SCENARIO='scenario_maze.png'; %<--- fil
        domainX=4; % length of domain (x-axis)
        xinc=100; %number of nodes across x-com
        dt=0.0025; %set set delta time [unit: s
        MI=2000; %number of time steps to perfo
        velyi=0; %y-component velocity of regic
                      %[velyi>0,velocity has vecto
        velxi=0.25; %x-component velocity of re
                      %[velxi>0,velocity has vecto
        %[if velxi=0.1 and velyi=-1, vector is
        dens=1; %density  [unit: kg/m^3] , wate
        mu=1/250; %0.001 %dynamic viscosity [kg

    %Poisson Pressure solver parameters!!!!!!!
        error=0.001; %set tolerance of error fc
        MAXIT=500; %maximum number of iteration
        MINIT=1; %mininum number of iterations
        % Note that: MINIT should be less than
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage
    ST=[100 100 500]; % FOR variables of dimens
                      % save variable data for
                      % in chuncks of files , e
                      % size matrix.........thi
                      %(increasing ST(3) will r
                      %(decreasing ST(3) will r
                      %[Files; openvar.m and sa
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%##############################################
```

Example outputs from settings on previous page

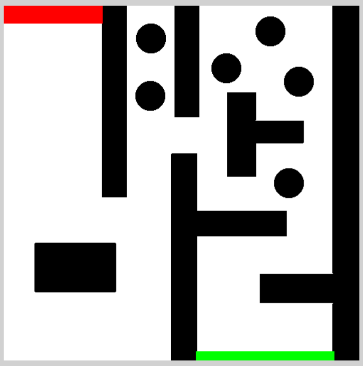






Example 7

This scenario simulates fluid flowing through a random maze.



Above: image that is imported into matlab

(.png file of resolution: 1000x1000)

Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

White = region of free fluid flow

Green = fluid outlet

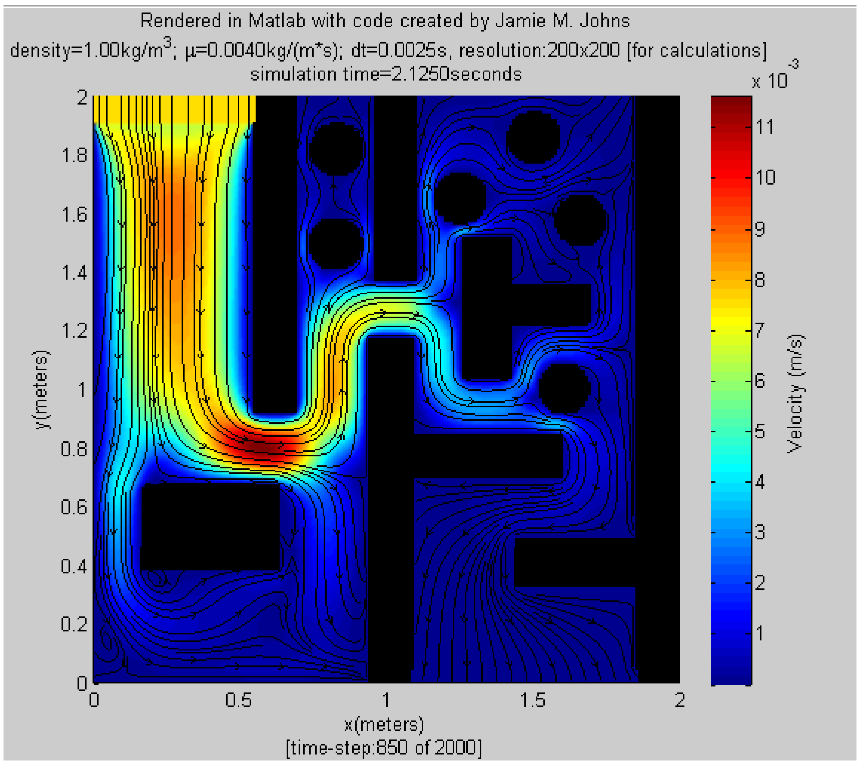
Black = region of no fluid flow ("solid wall")

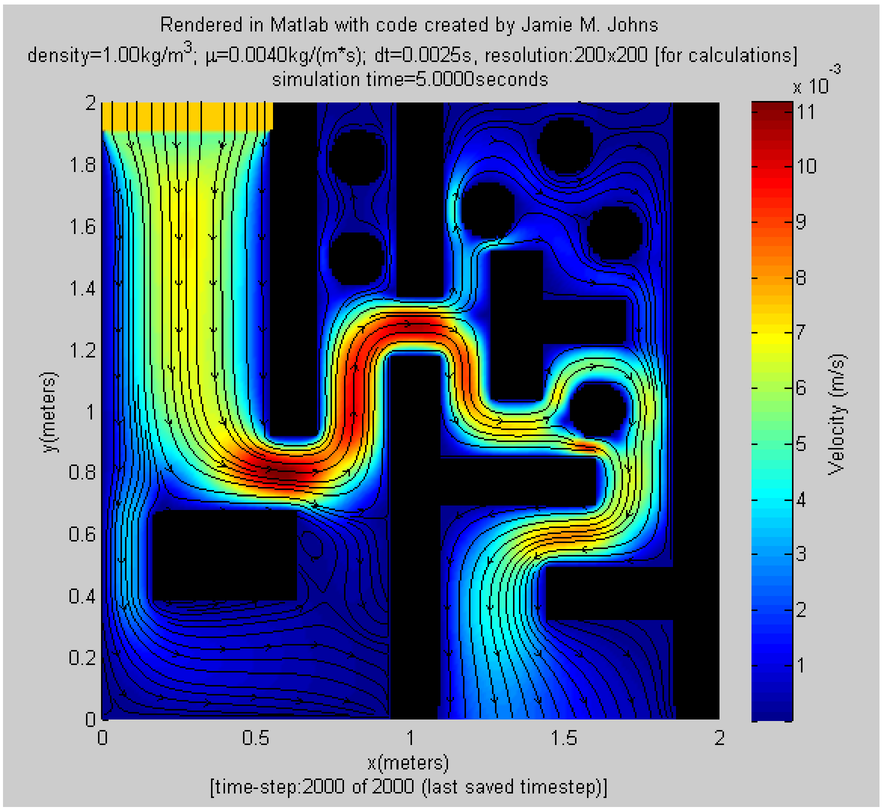
The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

Example settings

```
%Parameters for scenario (Modify these) ########
    %set information about domain of simulation@
        SCENARIO='scenario_maze_random.png'; %<-
        domainX=2; % length of domain (x-axis) [
        xinc=200; %number of nodes across x-comp
        dt=0.0025; %set set delta time [unit: se
        MI=2000; %number of time steps to perfor
        velyi=0.0075; %y-component velocity of r
                      %[velyi>0,velocity has vector
        velxi=0; %x-component velocity of region
                      %[velxi>0,velocity has vector
        %[if velxi=0.1 and velyi=-1, vector is =
        dens=1; %density  [unit: kg/m^3] , water
        mu=1/250; %0.001 %dynamic viscosity [kg/

    %Poisson Pressure solver parameters!!!!!!!!!
        error=0.001; %set tolerance of error for
        MAXIT=500; %maximum number of iterations
        MINIT=1; %mininum number of iterations a
        % Note that: MINIT should be less than M
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage
    ST=[100 100 500]; % FOR variables of dimensi
                      % save variable data for x
                      % in chuncks of files , ea
                      % size matrix.........this
                      %(increasing ST(3) will re
                      %(decreasing ST(3) will re
                      %[Files; openvar.m and sav
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%################################################
```

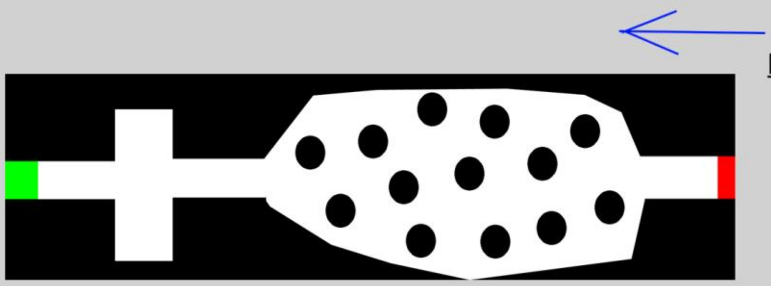
Example output from settings on previous page





Example 8

Another random fluid flow scenario.



Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

White = region of free fluid flow

Green = fluid outlet

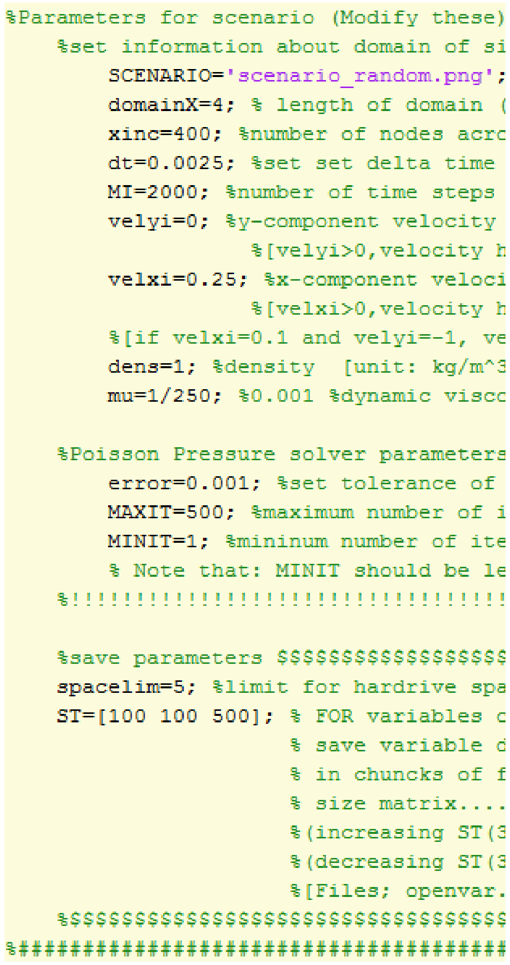
Black = region of no fluid flow ("solid wall")

Above: image that is imported into matlab

(.png file of resolution: 2000x500)

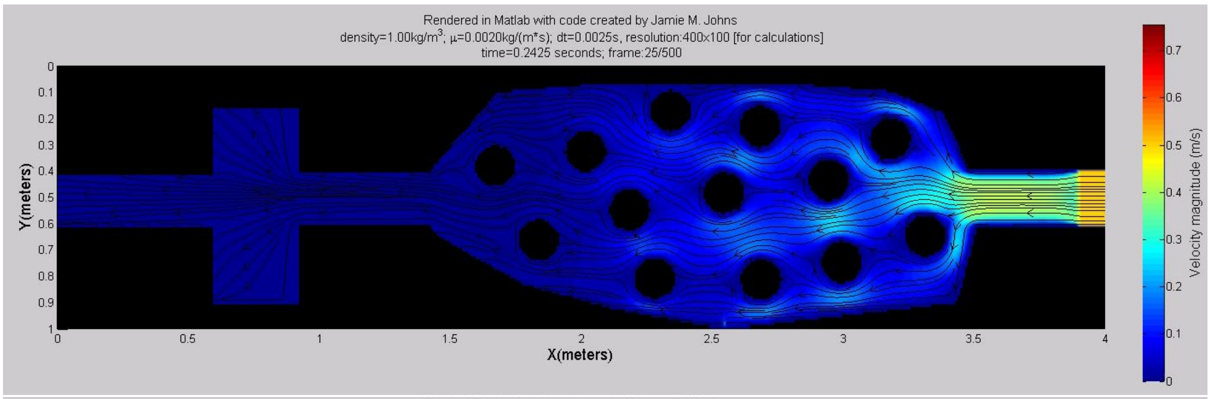
The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

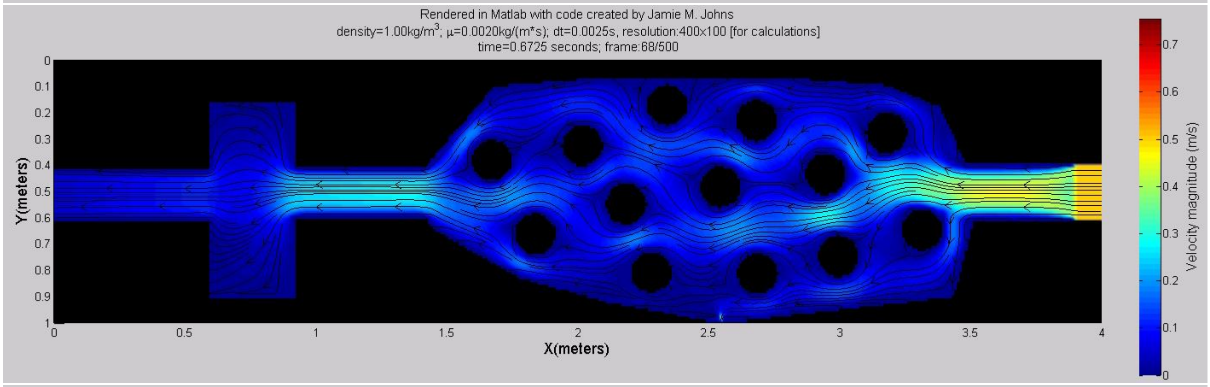
Example settings

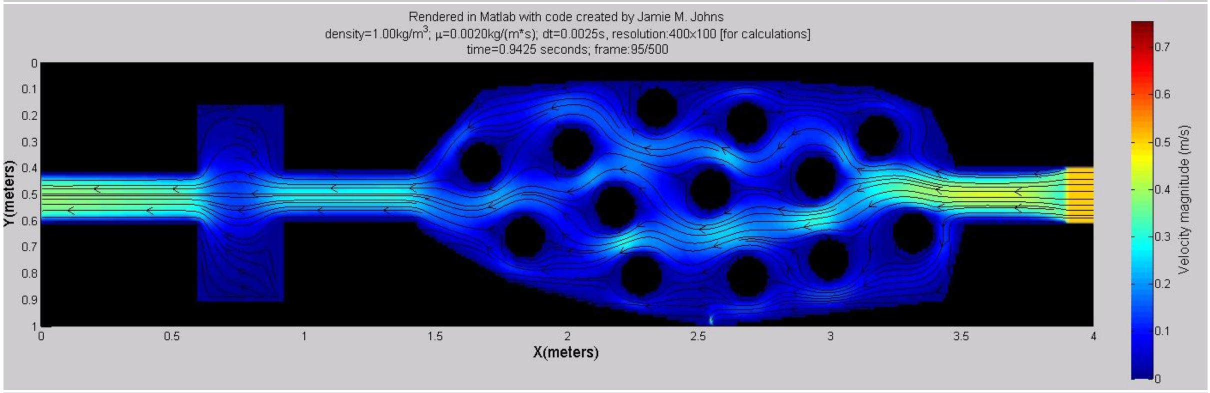


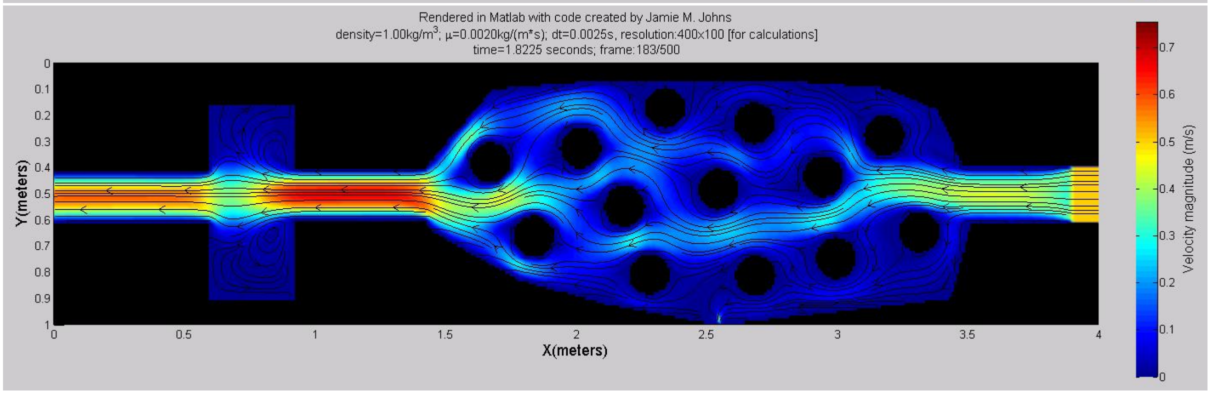
```
%Parameters for scenario (Modify these)
    %set information about domain of si
        SCENARIO='scenario_random.png';
        domainX=4; % length of domain (
        xinc=400; %number of nodes acro
        dt=0.0025; %set set delta time
        MI=2000; %number of time steps
        velyi=0; %y-component velocity
                   %[velyi>0,velocity h
        velxi=0.25; %x-component veloci
                   %[velxi>0,velocity h
        %[if velxi=0.1 and velyi=-1, ve
        dens=1; %density  [unit: kg/m^3
        mu=1/250; %0.001 %dynamic visco

    %Poisson Pressure solver parameters
        error=0.001; %set tolerance of
        MAXIT=500; %maximum number of i
        MINIT=1; %mininum number of ite
        % Note that: MINIT should be le
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive spa
    ST=[100 100 500]; % FOR variables c
                      % save variable d
                      % in chuncks of f
                      % size matrix....
                      %(increasing ST(3
                      %(decreasing ST(3
                      %[Files; openvar.
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%#######################################
```

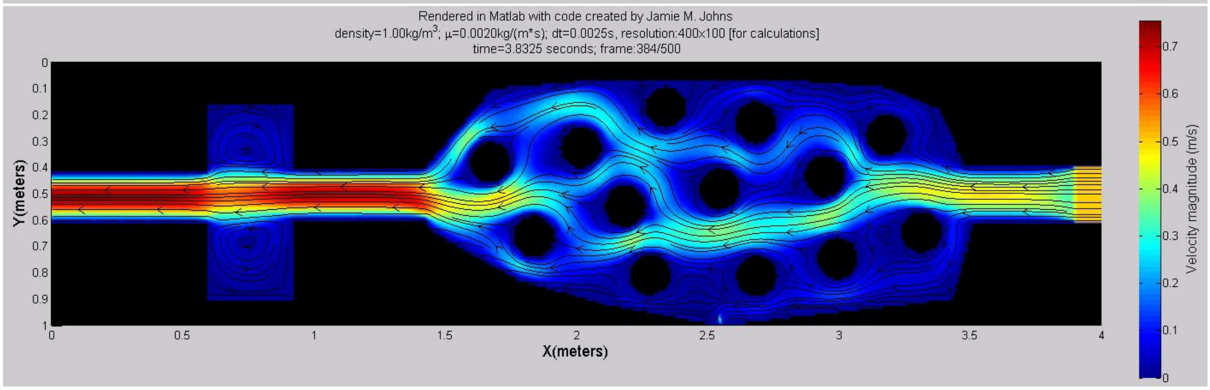
Example outputs from settings on previous page





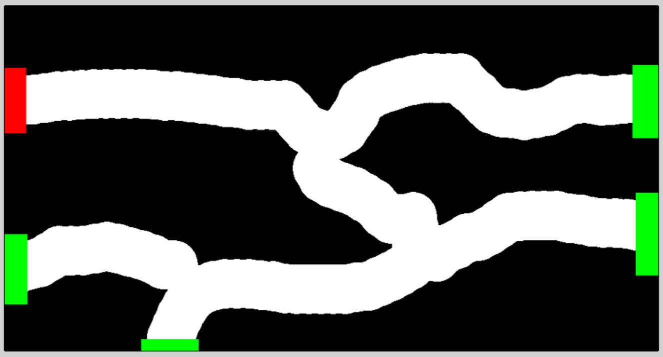






Example 9

Another random example simulating a creek/drain scenario.



Direction of the constant velocity specified in the code.

Pixel colour in imported diagram

Red=constant velocity region

White = region of free fluid flow

Green = fluid outlet

Black = region of no fluid flow ("solid wall")

above: image that is imported into matlab

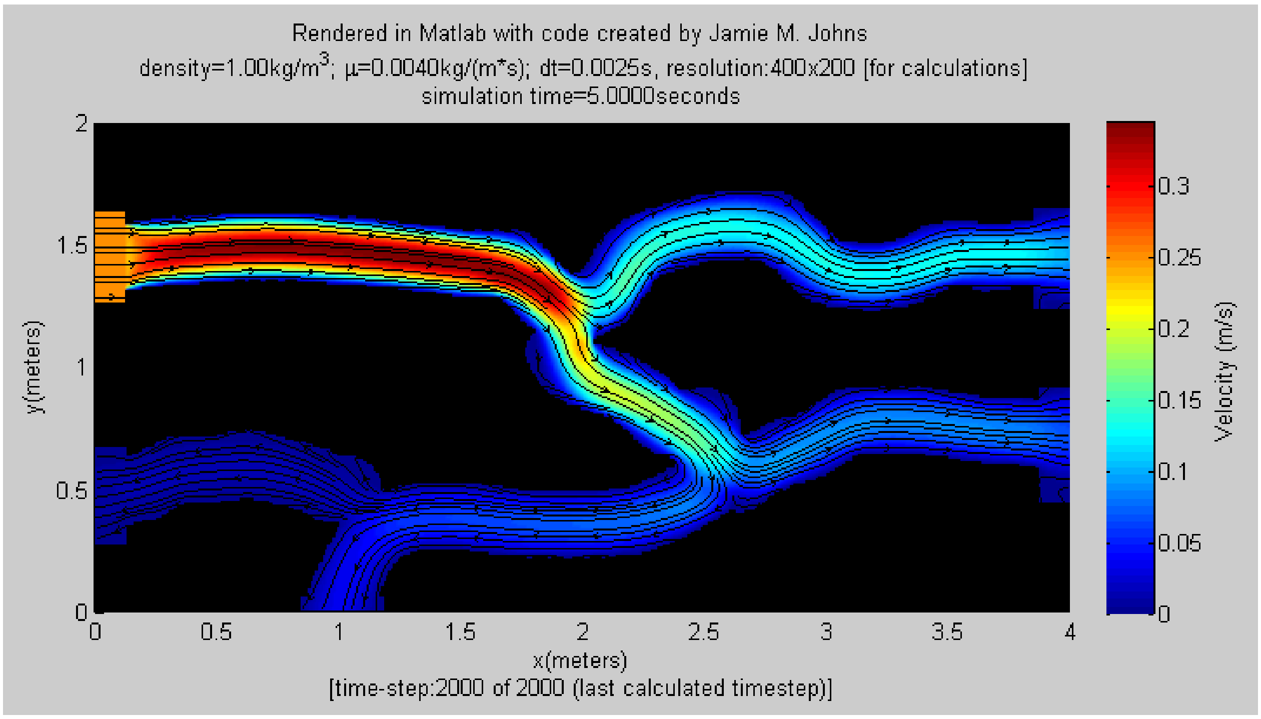
(.png file of resolution: 1000x500)

The edges of the imported image are handled as solid walls,except, for where an edge pixel (of image) is coloured green. Then, the edges at these locations will be handled as fluid outlet.

Example settings

```
%Parameters for scenario (Modify these) ####################
    %set information about domain of simulation@@@@@@@@@@@@@
        SCENARIO='scenario_creek.png'; %<--- file (image) th
        domainX=4; % length of domain (x-axis) [unit: meters
        xinc=400; %number of nodes across x-component of dom
        dt=0.0025; %set set delta time [unit: seconds]
        MI=2000; %number of time steps to perform calculatio
        velyi=0; %y-component velocity of region with consta
                   %[velyi>0,velocity has vector -y with mag
        velxi=0.25; %x-component velocity of region with con
                   %[velxi>0,velocity has vector +x with mag
        %[if velxi=0.1 and velyi=-1, vector is = 0.1[X]+(-1)
        dens=1; %density  [unit: kg/m^3] , water(pure)=1000 
        mu=1/250; %0.001 %dynamic viscosity [kg/(m*s)]

    %Poisson Pressure solver parameters!!!!!!!!!!!!!!!!!!!!
        error=0.001; %set tolerance of error for convergence
        MAXIT=500; %maximum number of iterations allowed for
        MINIT=1; %mininum number of iterations allowed for p
        % Note that: MINIT should be less than MAXIT
    %!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

    %save parameters $$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
    spacelim=5; %limit for hardrive space usage (in gigabyte
    ST=[100 100 500]; % FOR variables of dimensions of ST(1)
                      % save variable data for x and y compo
                      % in chuncks of files , each with ST(1
                      % size matrix.........this reduces mem
                      %(increasing ST(3) will reduce number 
                      %(decreasing ST(3) will reduce number 
                      %[Files; openvar.m and savevar.m are u
    %$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
%###########################################################
```

Example output from settings on previous page


Rendered in Matlab with code created by Jamie M. Johns
density=1.00kg/m^3; μ=0.0040kg/(m*s); dt=0.0025s, resolution:400x200 [for calculations]
simulation time=5.0000seconds
y(meters)
x(meters)
[time-step:2000 of 2000 (last calculated timestep)]
Velocity (m/s)