

DRAFT

A TWO-DIMENSIONAL VISCOUS-FLUID CONVECTION PROBLEM

Author: J. Horner, jhorner@cybermesa.com
Last modified: 6 August 2019/0935 US Central Time

Platform: *Mathematica* v12.0 ([1])
Dell Inspiron 1545
-- Intel(TM) Core2(TM) Quad CPU Q8200, clocked at 2.33 GHz
-- 8 GB physical memory
-- Windows 10

1.0 Introduction

This *Mathematica* notebook solves a two-dimensional (2D) viscous-fluid (heat-driven) convection problem. The code derives almost entirely from example in the Help documentation for “Solving Partial Differential Equations with Finite Elements” in [1]. (I made some minor changes to the code and augmented the discussion of the formulation as it appears in the original example.)

The notebook assumes acquaintance at the beginning graduate level with finite element methods for computational fluid dynamics ([2]), and acquaintance with *Mathematica* ([1], [3]).

The software in the notebook produces an animated visualization file, *hydro_visc_liquid_T1.avi*, renderable by Microsoft Media Manager, of the velocity and temperature fields.

1.1 Problem statement

Given the rectangular region with rectangular cutouts specified for “ Ω ” in Section 2 (and visualized in Section 4) containing a fluid that

- is incompressible and Newtonian
- has Prandtl number 7.4 and Rayleigh number $2 * 10^5$
- is initially at rest

```
-- has 1 K temperature difference between left and right faces of the region at t = 0
-- satisfies Dirichlet conditions at the boundary
```

compute the velocity, pressure, and temperature of the fluid from $t = 0$ to $t = 300$ s, interpolating the velocities u and v and the temperature T with second order and the pressure p with first order. Visualize the results in an AVI animation file (a video file renderable by Microsoft Media Manager).

1.2 Method

Apply the Boussinesq approximation (see [4], pp. 188-196 and [5]) to the problem, using an unstructured computational mesh ([6], esp. Section 1.4).

2.0 Begin executable code

Initialize the wall clock, in seconds since 1 January 1990.

```
In[68]:= begintime = AbsoluteTime[]
```

```
Out[68]= 3.7740729431732989 × 109
```

Provide access to the Mathematica finite element modeling package.

```
In[69]:= Needs["NDSolve`FEM`"] (* this is required *)
```

Specify a rectangular region with rectangular cutouts (see Section 4.0 for a visualization of the region boundary).

```
In[70]:= sizes = {length → 2, height → 1, thickness → 1/6, offset → 1/4};  
Ω = RegionDifference[RegionDifference[Rectangle[{0, 0}, {length, height}], Rectangle[  
    {length/2 - offset - thickness, height/3}, {length/2 - offset, height}]], Rectangle[  
    {length/2 + offset, 0}, {length/2 + offset + thickness, 1 - height/3}]] /. sizes;
```

Specify Prandtl and Rayleigh numbers for the fluid in the region and use them to compute the viscosity and thermal emissivity of the material. (The viscosity is comparable to that of castor oil.)

```
In[72]:= parameters = {ν → Sqrt[Pr / Ra], ε → 1 / Sqrt[Pr * Ra]} /. {Pr → 7.1, Ra → 2 * 105};
```

Define a viscous Navier–Stokes (N-S) system that is coupled to a heat equation. Use the Boussinesq

approximation.

```
In[73]:= op = {
  u(1,0,0)[t, x, y] + ∇{x,y} · (-v ∇{x,y} u[t, x, y]) +
  {u[t, x, y], v[t, x, y]} · ∇{x,y} u[t, x, y] + p(0,1,0)[t, x, y],
  v(1,0,0)[t, x, y] + ∇{x,y} · (-v ∇{x,y} v[t, x, y]) +
  {u[t, x, y], v[t, x, y]} · ∇{x,y} v[t, x, y] + p(0,0,1)[t, x, y] - T[t, x, y],
  u(0,1,0)[t, x, y] + v(0,0,1)[t, x, y],
  T(1,0,0)[t, x, y] + ∇{x,y} · (-ε ∇{x,y} T[t, x, y]) +
  {u[t, x, y], v[t, x, y]} · ∇{x,y} T[t, x, y]} /. parameters;
```

Define boundary conditions and initial conditions.

```
In[74]:= wall = DirichletCondition[{u[t, x, y] == 0, v[t, x, y] == 0}, True];
```

Define a pressure reference point.

```
In[75]:= reference = DirichletCondition[p[t, x, y] == 0, x == 0 && y == 0];
```

Define initial temperatures at the left and right walls of the region.

```
In[76]:= temperatures = {DirichletCondition[T[t, x, y] == 1, x == 0],
  DirichletCondition[T[t, x, y] == 0, x == length]};
```

Combine the wall, reference, and temperature conditions in a list.

```
In[77]:= bcs = {wall, reference, temperatures} /. sizes;
```

Specify that the system is initially at rest.

```
In[78]:= ic = {u[0, x, y] == 0, v[0, x, y] == 0, p[0, x, y] == 0, T[0, x, y] == 0};
```

3.0 Solve the system

Use `NDSolveValue` to solve the fluid dynamics equations (“op”, above) for $t = 0$ to $t = 300$ time units over the mesh. Monitor the time-progress of the solution and the total time it takes to solve the PDE, interpolating the velocities u and v and the temperature T with second order and the pressure p with first order.

```

In[79]:= Monitor[AbsoluteTiming[
  {xVel, yVel, pressure, temperature} =
    NDSolveValue[{op == {0, 0, 0, 0}, bcs, ic}, {u, v, p, T}, {x, y} ∈ Ω, {t, 0, 300},
    Method → {
      "PDEDiscretization" → {"MethodOfLines",
        "SpatialDiscretization" → {"FiniteElement", "MeshOptions" → {"MaxCellMeasure" →
          0.00125}, "InterpolationOrder" → {u → 2, v → 2, p → 1, T → 2}}}]
    , EvaluationMonitor → (currentTime = Row[{"t = ", CForm[t]}])];
  ], currentTime]
Out[79]:= {125.38, Null}

```

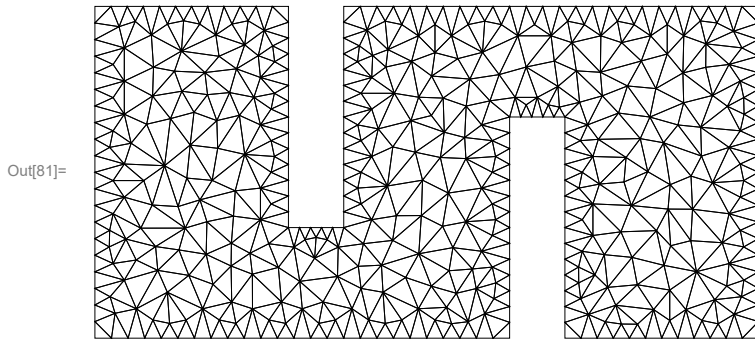
4.0 Visualize the results

Compute the mesh boundary.

```
In[80]:= bmesh = ToBoundaryMesh[xVel["ElementMesh"]];
```

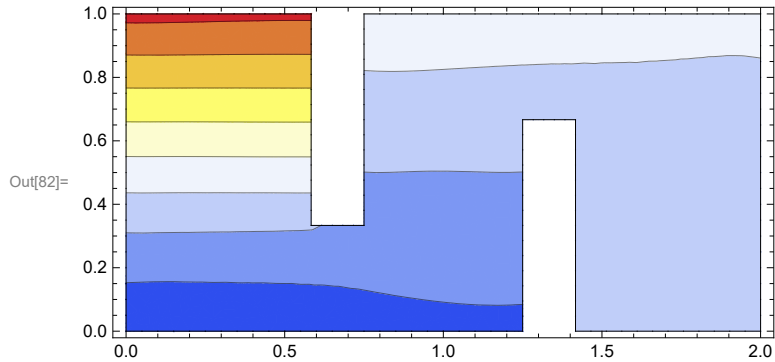
Visualize the full mesh.

```
In[81]:= ToElementMesh[bmesh] ["Wireframe"]
```



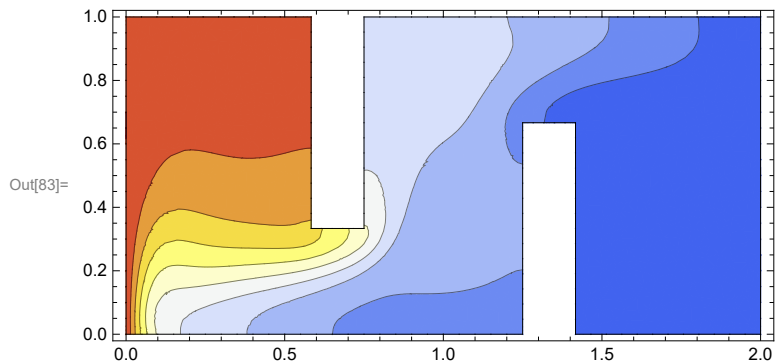
Visualize the pressure distribution in the region at $t = 150$.

```
In[82]:= Show[ContourPlot[pressure[150, x, y], {x, y} ∈ pressure["ElementMesh"], ... +],  
  bmesh["Wireframe"]]
```



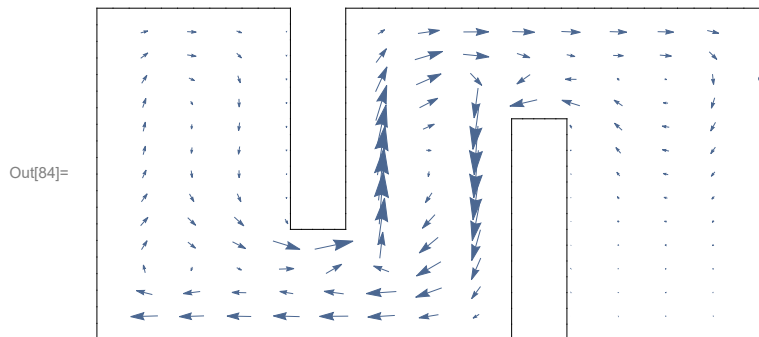
Visualize the temperature distribution in the region at $t = 150$.

```
In[83]:= Show[ContourPlot[temperature[150, x, y],  
  {x, y} ∈ temperature["ElementMesh"], ... +], bmesh["Wireframe"]]
```



Visualize the velocity field at $t = 150$.

```
In[84]:= Show[bmesh["Wireframe"],  
  VectorPlot[{xVel[150, x, y], yVel[150, x, y]}, {x, y} ∈ xVel["ElementMesh"], ... +]]
```



Compute the temperature contour-plot and velocity stream-plot, superposed, as video frames every 2.5 time units. This may takes several minutes to execute.

```
In[85]:= frames = Table[Show[
  bmesh["Wireframe"],
  ContourPlot[temperature[t, x, y], {x, y} ∈ temperature["ElementMesh"], ... +],
  StreamPlot[{xVel[t, x, y], yVel[t, x, y]}, {x, y} ∈ xVel["ElementMesh"], ... +]
], {t, 0, 300, 2.5}];
```

Export the video frames in AVI format (120 video frames at 15 frames per second) to a file.

```
In[86]:= Export["C:\\Hydrodynamics\\hydro_visc_liquid_T1.avi", frames];
```

Uncomment the following instruction to implement an animation of “frames” that is visualizable within this notebook. (The instruction is commented out because it interferes with conversion of this notebook to a PDF file.)

```
In[87]:= (* ListAnimate[frames, SaveDefinitions→True] *)
```

Compute the elapsed wall-clock (execution) time in seconds.

```
In[88]:= endtime = AbsoluteTime[]
```

```
Out[88]= 3.7740735184233008 × 109
```

```
In[89]:= elapsedtime = endtime - begintime
```

```
Out[89]= 575.2500019
```

5.0 REFERENCES

[1] Wolfram Research. (2019). *Mathematica* v12.0 Home Edition. <http://www.wolfram.com/mathematica-home-edition/>. Accessed 23 July 2019.

[2] Kuzmin D and Hämäläinen J. (2014). *Finite Element Methods for Computational Fluid Dynamics: A Practical Guide*. SIAM.

[3] Napolitano J. (2018). *A Mathematica Primer for Physicists*. CRC Press.

[4] Tritton DJ. (1988). *Physical Fluid Dynamics*. Second Edition. Oxford.

[5] Boussinesq MJ. (1897). *Théorie de l'écoulement tourbillonnant et tumultueux des liquides dans les lits rectilignes a grande section*. Paris: Gauthier-Villars et Fils. Available online at <https://archive.org/details/thbeoriedelbeco01bousrich/page/n8>. Accessed 5 August 2019.

[6] Thompson JF, Soni BK, and Weatherill NP (eds.). (1999). *Handbook of Grid Generation*. CRC Press.