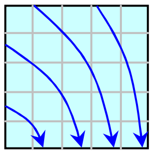
Fluid Mechanics Dr. Pogo

Handout: CFD for “Potential Flow” using Streamfunctions

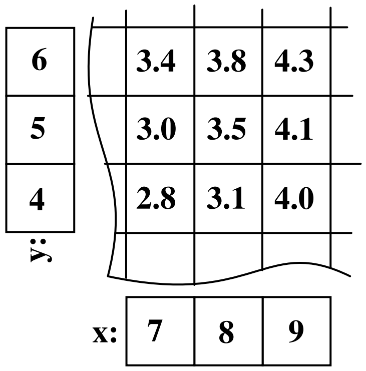
Overview

Recall that for 2D irrotational, steady, incompressible flow, the momentum equation reduces to $\nabla^2\psi = 0$. Let’s imagine a region of fluid where we want to solve this equation computationally. A small portion of some such flow is shown here. We’ll discretize this region into small rectangular chunks; for each small chunk, we’ll assume that ψ is fairly constant. For the flow field shown, a fragment of this grid might look as shown in the second picture. The numbers indicate values for ψ. Of course, in every direction, the flow field might extend further than the 3×3 portion shown here.



How can we check that this region satisfies Laplace’s equation ($\nabla^2\psi = 0$)? We don’t have an analytic function for ψ, so it is difficult to take a derivative. Instead, we’ll have to *approximate* the derivatives we need:

$$\nabla^2\psi = \frac{\partial^2\psi}{\partial x^2} + \frac{\partial^2\psi}{\partial y^2} = 0.$$



Each derivative can be approximated by: $\left.\dfrac{\partial^2\psi}{\partial x^2}\right|_{i,j} \approx \dfrac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{\Delta x^2}$.

So, the Laplacian becomes: $\nabla^2\psi_{i,j} \approx \dfrac{\psi_{i-1,j} - 2\psi_{i,j} + \psi_{i+1,j}}{\Delta x^2} + \dfrac{\psi_{i,j-1} - 2\psi_{i,j} + \psi_{i,j+1}}{\Delta y^2} = 0$.

Here, the i and j subscripts refer to grid numbers. For example, $\psi_{8,5} = 3.5$.

Fortunately, for our grid, $\Delta x = \Delta y$. Solving for $\psi_{i,j}$: $\boxed{\psi_{i,j} = \dfrac{\psi_{i+1,j} + \psi_{i,j-1} + \psi_{i-1,j} + \psi_{i,j+1}}{4}}$

In other words, if the value of ψ in each cell is equal to the average of the 4 directly neighboring cells, then $\nabla^2\psi \approx 0$. So as a check, we should see if 3.5 is the average of 3.8, 3.0, 4.1, and 3.4 (it is).

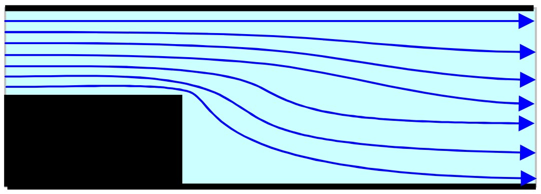
Boundary Conditions

In practice, we are often more interested in going the other way: we want to start out with some boundary conditions for ψ, and determine the value of ψ at every point inside the region of interest. Setting the boundary conditions is the hard part. Let’s remind ourselves of what ψ means: it tells us the components of velocity: $\dfrac{\partial\psi}{\partial y} = u$, and $-\dfrac{\partial\psi}{\partial x} = v$.

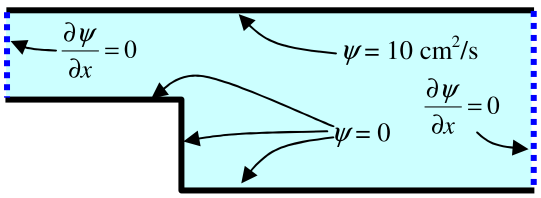
So, if we set $\dfrac{\partial\psi}{\partial x} = 0$ along some boundary, we are saying that the flow is all in the x direction. Furthermore, we know that the flow between any two streamlines ψ_1 and ψ_2 is $Q = w(\psi_2 - \psi_1)$, where w is the thickness of the flow (into the page). We can use this to set some boundaries of the flow as exactly equal to some value. Remember that as with all potential flows, since our model is frictionless, our results will violate the no-slip condition.

Example

Consider a flow of water in a channel that suddenly expands. The height of the entry channel might be 1m, and the exit channel is 2 m tall. Horizontally, the total region shown might be 5 m long, with an obstruction of 2 m length. Q is given as 100 cm^3/s, and the channel is w = 10 cm wide into the page. We expect the results to look somewhat as shown.



Our region of flow seems to have 6 total boundaries. We are always free to pick any arbitrary value of ψ as a starting point, since only *differences* in ψ have any physical meaning. For convenience, we’ll choose $\psi = 0$ every along the bottom of the flow (3 surfaces!). These 3 surfaces must all have the same ψ because fluid that starts out adjacent to this surface can’t leave it later on (without crossing another streamline, which is forbidden). After this value is picked, the very top surface must therefore be $\psi = Q/w = 10$ cm^2/s.

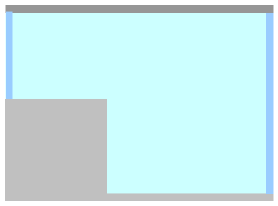


$\dfrac{\partial\psi}{\partial x} = 0$
$\psi = 10$ cm^2/s
$\dfrac{\partial\psi}{\partial x} = 0$
$\psi = 0$

We have a few options for how we set the boundary conditions on the remaining surfaces. We might choose to pick a range of values between $\psi = 0$ and $\psi = 10$ cm^2/s, and distribute them evenly across each of these two boundaries. However, it is probably more appropriate to declare that these flows are *horizontal*, as shown on the figure.

Getting Stuff to work in Excel (and Mathematica)

1. Select the space to the left of the “A” label (and above the “1” label). Change the font size to 8. While everything is still selected, grab the line next to any label letter, and shrink the width sideways so that each cell looks like a square. Every cell will contain a value for ψ.
2. Grab some cells (the quantity is unimportant, perhaps 50 wide $\times$ 25 tall). When doing “real” calculations, you’d need to know the dimensions of each cell (e.g., 1 cm^2). Use the paintbucket to color them light blue. Then, select some cells in the lower left corner (perhaps 20 wide by 17 tall). You should use less than half the total width, but more than one quarter of the total width. The height is less important. Color these cells light gray.



3. Color all the light blue cells on the bottom edge light gray. Color all the light blue cells in the top row a slightly darker gray. Color all the light blue cells along the left and right edges dark blue.
4. Set some boundary conditions for our PDE: in each light gray cell, type a 0. In each dark gray cell, type a 10. This number has units, so when doing “real” calculations, you’d need to make sure you use the correct value!
5. In each darker blue cell on the left, enter an equation setting it equal to the cell to its right. For each darker blue cell on the right, type an equation setting it equal to the cell to its left. These boundary conditions force the inlet and outlet flows to be horizontal.
6. Pick any light blue cell. Enter the equation for the PDE: this cell’s value is ¼(sum of the neighboring 4 cells).
7. Drag this equation around until it appears in EVERY light blue cell. Ignore all of Excel’s whining about “circular references”, and close any on-screen complaints.
8. Tell Excel to use iterative methods to solve the problem: go to *File/options/formulas*. Turn on both the iterations box and the “manual” button. Set the # of iterations to 200. Go back to the main screen.
9. Each type you press F9, Excel will run 200 iterations of every cell. You’re done when the values stop changing!
10. Make a surface plot: select all the colored cells, go to “recommended charts/all charts/surface/contour”.
11. Delete the legend, then resize the plot more appropriately. Click the plot, then click the green “Plus” symbol near the upper right corner. Hover to the right of the word “axis” until a gray triangle appears, then click it. Uncheck “primary vertical”, then recheck it to turn it back on. Some goofy looking numbers have now appeared in a stack near the lower left corner of the plot.
12. Double click the new numbers. On the sidebar, use the “axis options” to make sure the toolbar for “vertical (value) axis” is visible (a little green bar chart), then click it. Set min =0, max = 10, and major = 0.5. This will make our plot have 20 contours.
13. On the plot, click the axis labels that have “series” in them. On the sidebar, check “series in reverse order”.
14. This plot is about the best Excel can do. Let’s take the data and try to shove it into Mathematica. Select all the colored cells, and press “control C” (for “copy all”).
15. In Mathematica, enter the following two lines:

```
ψ = ImportString["pasted", "TSV"];
ListContourPlot[ψ, Contours -> 20, ColorFunction -> "Rainbow", RegionFunction -> Function[{x, y,
z}, Or[x > 19, y > -16 ]], ScalingFunctions -> {None, "Reverse"},AspectRatio->True]
```

16. Select or highlight the word “pasted” without including the quotes. Press “control V” to paste our copied data from Excel. Accept whatever defaults Mathematic a whines about. Your .nb is still only two lines long, but it looks like a thousand!
17. Change the constants 19 and -16 to match the height and width of the corner you actually created in step 2.
18. Ok, it’s true: plots of ψ should never have color! You might change the “ColorFunction” part to this instead: ContourShading -> None. However, this makes it harder to see where the corner actually was. You might make a separate “plot” that has a black corner, and then combine both of them with a “Show”.