Potential Flow and Related Problems

Stephen Neethling

Potential Flow

- Often used for the specific case of inviscid (zero viscosity), incompressible fluid flow

- In this lecture I will be using potential flow to describe a wider number of problems which can be modelled using Laplace’s equation

- Will look at numerical solutions of Laplace’s equation
  - Analytical solutions do exist, but only in the form of infinite series

Learning Objectives

- Derive the potential flow equation
  - Laplace's equation for constant resistance
- Be able to recognise a range of physical systems that obey potential flow
- Show that the Navier-Stokes equation can be approximated as a potential flow problem under certain circumstance
  - Including deriving a vorticity formulation of the Navier-Stokes equation
- Demonstrate the stream function formulation for solving incompressible flow problems
  - Demonstrate with potential flow problems
- Demonstrate numerical solutions for Laplace’s equation

What is potential flow?

- Flow is proportional to the gradient of a potential:

$$\mathbf{v} = -k\nabla\varphi$$

- Flow continuity holds:

$$\nabla \cdot \mathbf{v} = 0$$

- If the proportionality is constant this results in Laplace’s Equation:

$$\nabla^2\varphi = 0$$

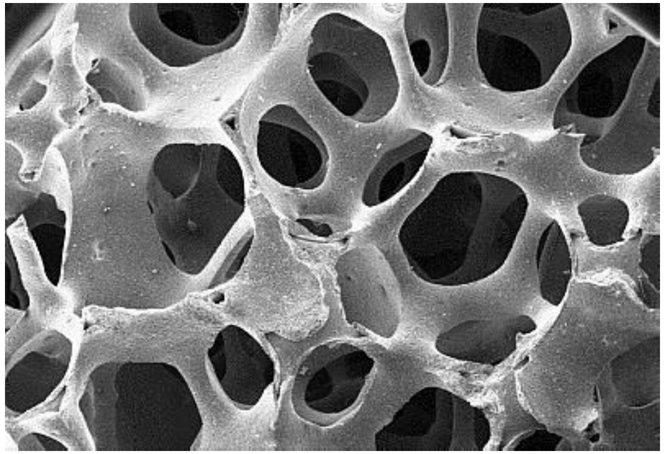
Where potential flow is a good approximation

One type of situation where potential flow is a good approximation

- The potential exerts a “force” on the quantity being conserved
  - Electrical current, fluid volume…
- The resistance to the flow is proportional to the flow-rate of the conserved quantity

- Good approximation for a number systems
  - Heat flow
  - Electrical current
  - Saturated flow in porous media

Flow in Porous Media



- Wide range of systems
  - Packed bed catalytic reactors
  - Water flow in aquifers
  - Petroleum reservoirs
  - Porous electrodes in flow batteries and fuel cells
  - …

Flow in Porous Media

- Often characterised by a permeability, k, with viscosity as a separate term in order to separate the porous media and the fluid effect:

$$\mathbf{v} = -\frac{k}{\mu}\nabla P$$

- Note that this velocity is the superficial velocity (volumetric flowrate per area of porous media) rather then the actual average velocity of the fluid
  - The porosity of the media is the factor by which the superficial velocity will be lower than the average fluid velocity
- More complicated for multi-phase flow in porous media
  - Introduction of relative permeabities that are a function of volume fraction of that phase in the pore space
  - Capillarity (forces due to surface tension) can further complicate the situation

Random motion

- Random motion can also lead to potential flow
- Diffusion is a good example
  - Thermal motion of molecules in a solvent leads to molecules of the solute being randomly nudged – **Brownian motion**
  - At the microscopic scale heat flow follows a similar process as the kinetic energy of the particles is randomly transferred to neighbouring particles
- If particle motion is random, the number of particles leaving a small volume in a given time will be proportional to the number of particles in the volume
  - The number of particles entering will be proportional to the number of particles in the neighbouring regions
  - This will result in a net flow down a concentration gradient

Diffusion through a random walk

- If we assume that Brownian motion causes a particle of the solute to take a random step of size Δx in time Δt, this will result in the following macroscopic diffusion coefficient:
  - Δx is the Root Mean Square (RMS) of the step size if the step size varies

$$D = \frac{\Delta x^2}{2\Delta t}$$

- The macroscopic flux can then be expressed as a function of the diffusion coefficient and the gradient of the concentration:

$$\boldsymbol{F} = -D\nabla C$$

Implications of Potential Flow

- The flux in pure potential flow is always irrotational
- Flow is irrotational (curl of the flux vector is zero):

$$\nabla \times \mathbf{v} = 0$$

- Prove?

$$\nabla \times \mathbf{v} = \begin{pmatrix} \frac{\partial v_z}{\partial y} - \frac{\partial v_y}{\partial z} \\ \frac{\partial v_x}{\partial z} - \frac{\partial v_z}{\partial x} \\ \frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \end{pmatrix}$$

- Reverse is also true: Irrotational flows can be written as potential flows

Calculation

- Demonstrate that potential flows are irrotational

What is vorticity?

- Potential flow implies irrotationality, which means that it has a vorticity of zero

  $$\omega = \nabla \times \mathbf{v}$$

- Vorticity, ω, is a measure of the rotation of the flow
  - This is true, but can be misleading – fluid that has a curving flow path can have a zero vorticity, while fluid going in straight line can have a non-zero vorticity
  - Vorticity is a measure of the rotation of a small parcel of fluid – if you put a small twig in the flow will it rotate (the average path it follows is immaterial to the vorticity)?
  - In a 3D flow, the axis of the rotation is in the direction that the vorticity vector points and the length of the vector is proportional to how fast it rotates.

Incompressible Inviscid Flow

- Why is incompressible inviscid flow a potential flow problem?

Navier-Stokes Equations for Incompressible Newtonian Flow:

  - You have already derived the NS equation (more detail on it later in the course)
  - Need to know what each term represents

Flow of momentum (Inertial force)

Viscous force

Momentum balance: $$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \mathbf{u} \cdot \nabla \mathbf{u} = -\nabla P + \mu \nabla^2 \mathbf{u} + \rho\, \mathbf{g}$$

Rate of change of momentum

Pressure force

Body force

Incompressible continuity: $$\nabla \cdot \mathbf{u} = 0$$

$$\mathbf{u} \cdot \nabla \mathbf{u} = (\mathbf{u} \cdot \nabla)\mathbf{u} = \mathbf{u} \cdot (\nabla \mathbf{u}) = \begin{pmatrix} u_x \frac{\partial u_x}{\partial x} + u_y \frac{\partial u_x}{\partial y} + u_z \frac{\partial u_x}{\partial z} \\ u_x \frac{\partial u_y}{\partial x} + u_y \frac{\partial u_y}{\partial y} + u_z \frac{\partial u_y}{\partial z} \\ u_x \frac{\partial u_z}{\partial x} + u_y \frac{\partial u_z}{\partial y} + u_z \frac{\partial u_z}{\partial z} \end{pmatrix}$$

Incompressible Inviscid Flow

We can obtain the vorticity equation by taking the curl ($\nabla \times$) of the Navier-Stokes equation. Using $\boldsymbol{\omega} = \nabla \times \mathbf{u}$ as the variable for the vorticity:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \frac{\mu}{\rho}\nabla^2 \boldsymbol{\omega} + \nabla \times \mathbf{g}$$

Note that the pressure does not appear in this equation if the density is constant (incompressible flow) as the curl of the gradient of a scalar is always zero.

Try and derive this equation for yourself

You will also need to use the incompressible assumption: $\nabla \cdot \mathbf{u} = \mathbf{0}$.

Useful identities for deriving vorticity equation:

$$\nabla \times (\mathbf{a} + \mathbf{b}) = \nabla \times \mathbf{a} + \nabla \times \mathbf{b}$$

$$\nabla \times (\nabla^2 \mathbf{a}) = \nabla^2 (\nabla \times \mathbf{a})$$

$$\nabla \times (\nabla a) = \mathbf{0}$$

$$\nabla \times \frac{\partial \mathbf{a}}{\partial t} = \frac{\partial}{\partial t}(\nabla \times \mathbf{a})$$

$$\nabla \times (\mathbf{a} \times \mathbf{b}) = -\mathbf{b}(\nabla \cdot \mathbf{a}) + (\mathbf{b} \cdot \nabla)\mathbf{a} - (\mathbf{a} \cdot \nabla)\mathbf{b}$$

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla\left(\frac{1}{2}\mathbf{u} \cdot \mathbf{u}\right) - \mathbf{u} \times \boldsymbol{\omega}$$

This implies that:

$$\nabla \times \big((\mathbf{u} \cdot \nabla)\mathbf{u}\big) = (\mathbf{u} \cdot \nabla)\boldsymbol{\omega} + (\nabla \cdot \mathbf{u})\boldsymbol{\omega} - (\boldsymbol{\omega} \cdot \nabla)\mathbf{u}$$

Derive the NS vorticity formulation

- Show that: $$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \frac{\mu}{\rho} \nabla^2 \boldsymbol{\omega} + \nabla \times \mathbf{g}$$

Incompressible Inviscid Flow

If we further assume that the flow is inviscid (zero μ) and that there is a constant (or conservative, $\nabla \times \mathbf{g} = 0$) body force:

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u}$$

The LHS of this equation represents the evolution of the vorticity of a parcel of fluid. This implies that if the vorticity starts with a zero value it will remain zero for all positions and all times.

If we therefore assume that our flow is initially irrotational (zero vorticity), it will remain irrotational if it is incompressible and inviscid.

Incompressible Inviscid Flow

- Side note:

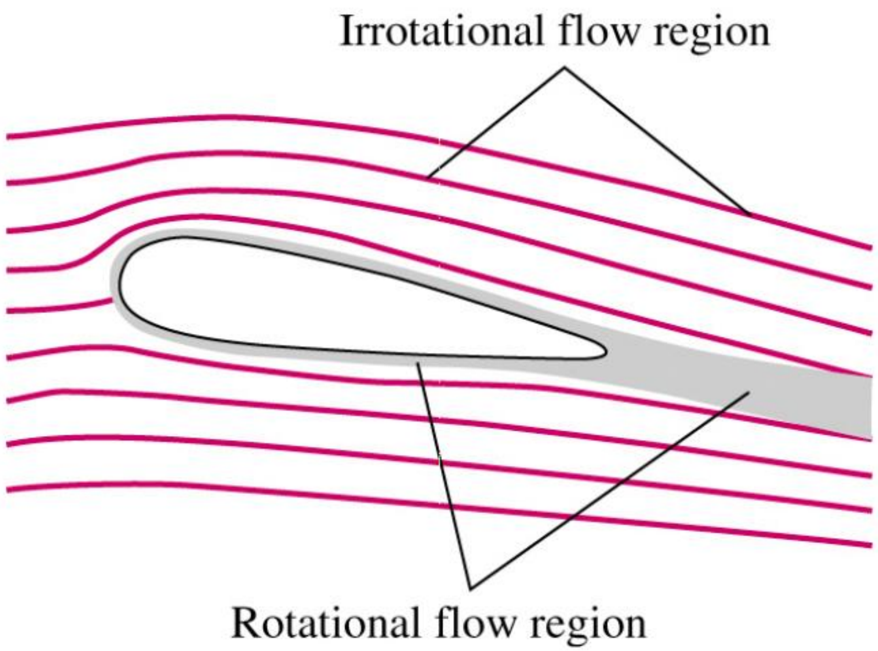
Zero vorticity is also a solution to the equation with viscosity

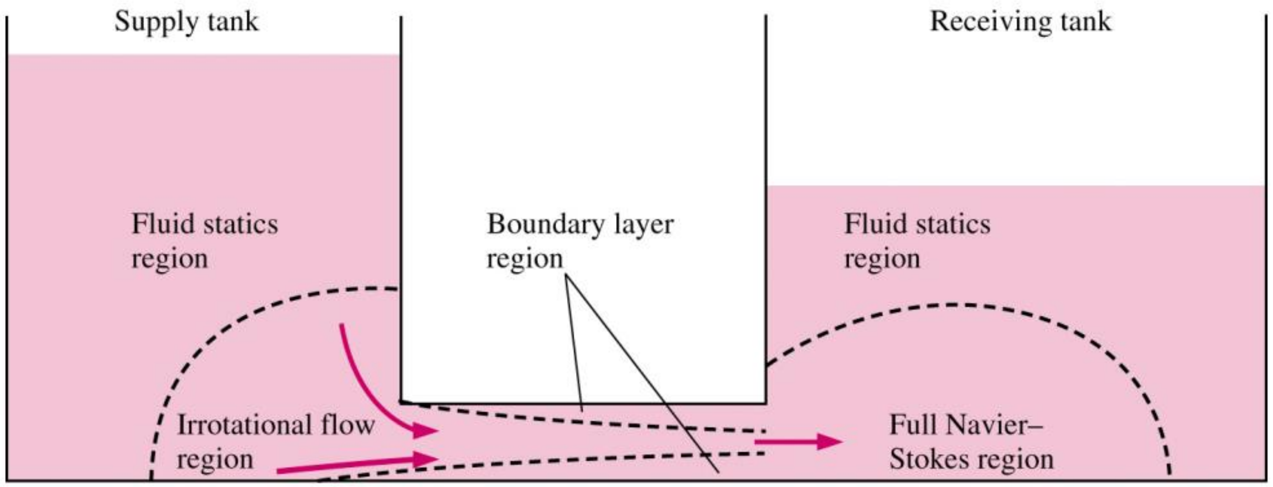
$$\frac{\partial \boldsymbol{\omega}}{\partial t} + \mathbf{u} \cdot \nabla \boldsymbol{\omega} = (\boldsymbol{\omega} \cdot \nabla)\mathbf{u} + \frac{\mu}{\rho} \nabla^2 \boldsymbol{\omega}$$

…,but it is incompatible with a shear stress at the boundaries. A shear stress at the boundary will generate vorticity at the boundary. The viscous term will then cause the vorticity to “diffuse” into the interior.

Incompressible Inviscid Flow

- Because vorticity diffuses in from boundaries, systems can have regions where potential flow is a good approximation and regions where it is not:





Incompressible Inviscid Flow

From our earlier discussion, irrotational flows can be described using a potential:

$$\mathbf{u} = -\nabla\varphi$$

As $\nabla \cdot \mathbf{u} = \mathbf{0}$, this further implies that

$$\nabla^2\varphi = 0$$

Incompressible Inviscid Flow

The potential described previously is NOT the same as the fluid pressure. To convert between them we need to use the Navier-Stokes Equation.

Navier-Stokes Equations for Incompressible Newtonian Flow:

$$\cancel{\frac{\partial \mathbf{u}}{\partial t}} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{\nabla P}{\rho} + \cancel{\frac{\mu}{\rho} \nabla^2 \mathbf{u}} + \mathbf{g}$$

$$\nabla \cdot \mathbf{u} = 0$$

Assuming steady-state and inviscid

Incompressible Inviscid Flow

Left with

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = -\frac{\nabla P}{\rho} + \mathbf{g}$$

$$\nabla \cdot \mathbf{u} = 0$$

We can also use the identity from earlier:

$$(\mathbf{u} \cdot \nabla)\mathbf{u} = \nabla\left(\frac{1}{2}\mathbf{u} \cdot \mathbf{u}\right) - \mathbf{u} \times \boldsymbol{\omega}$$

Since $\boldsymbol{\omega}$ is zero, this means that for an incompressible fluid (constant ρ):

$$\nabla\left(\frac{1}{2}\mathbf{u} \cdot \mathbf{u}\right) = \nabla\left(-\frac{P}{\rho} + \mathbf{g} \cdot \mathbf{x}\right)$$ Where $\mathbf{x}$ is the location vector

Incompressible Inviscid Flow

Since $\mathbf{u} = -\nabla\varphi$ based on our previous derivation:

$$\nabla\left(\frac{|\nabla\varphi|^2}{2}\right) = \nabla\left(\frac{P}{\rho} - \mathbf{g}\cdot\mathbf{x}\right)$$

This means that (less a constant of integration):

$$P = \rho\left(\frac{|\nabla\varphi|^2}{2} + \mathbf{g}\cdot\mathbf{x}\right)$$

Potential should be solved for first and the pressure subsequently calculated even though they are not the same as one another

Solving Potential Flow Problems

- While potential flow problems have a variety of physics that can be represented, the underlying equation that is being solved is always Laplace's Equation

$$\nabla^2 \varphi = 0$$

- There is no time dependency in this equation and therefore its solution takes the form of a Boundary Value Problem

- We therefore need to specify all boundaries

Boundaries Conditions

- Two linear boundary types are the specification of the potential at the boundary or the flux through the boundary

- Potential specification is a Dirichlet Boundary condition in potential:

$$\varphi = f(\mathbf{x})$$

- Flux specification is a Neumann boundary condition in potential:

$$\mathbf{v} \cdot \hat{\mathbf{n}} = -k\nabla\varphi \cdot \hat{\mathbf{n}} = F(\mathbf{x})$$

Where $\hat{\mathbf{n}}$ is the unit normal to the boundary and F is the scalar flux through the boundary (in the direction of $\hat{\mathbf{n}}$)

Stream Function Formulation

- Dirichlet boundary conditions are easier to implement than Neumann boundaries and typically result in quicker convergence when using a numerical solution scheme

- A stream function formulation allow flux boundary conditions to be specified using Dirichlet boundary conditions
  - Still doesn’t allow a mixture of potential and flux boundaries, which would still require a combination of Dirichlet and Neumann boundaries
    - We solve problems of this type in the next lecture

Stream Functions

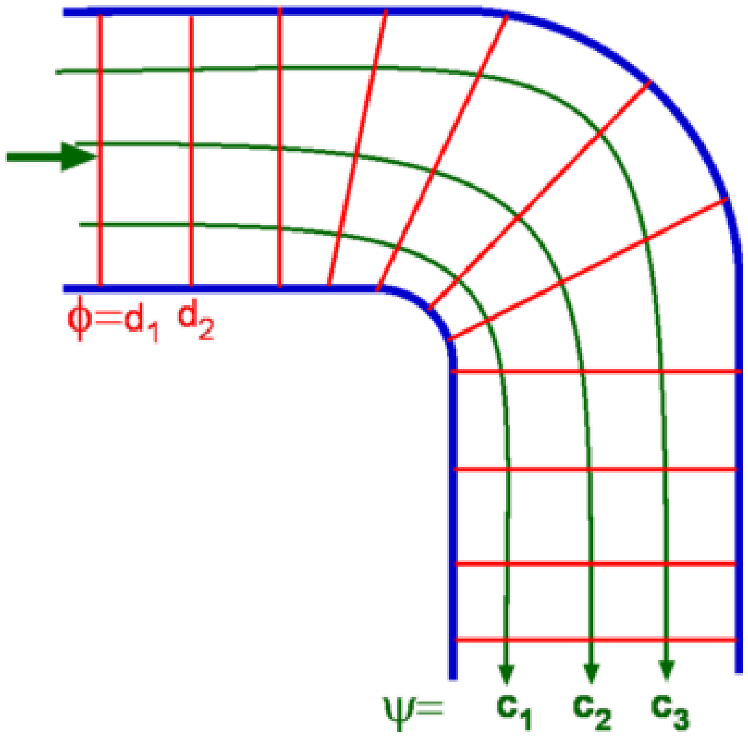
- Can be used to represent 2D steady state systems that exhibit continuity
- A stream function is a scalar field where flow follows a constant value of the function
  - In other words a constant value of the stream function represents a streamline
- Stream functions are just one way to solve potential flow problems and they only work in 2D
  - Can also be used in the analysis or solution of other flow problems where continuity of flux holds
- Make specifying flux boundary conditions easier
  - For flux boundary conditions turns a the specification of a potential gradient at the boundary into the specification of a stream function value
  - i.e. turns a Neumann boundary condition (harder to solve) into a Dirichlet boundary condition (easier to solve)

Stream Functions

Other properties of stream functions

- Differences in value of the stream function are equal to the flowrate of the conserved quantity between the streamlines represented by those stream function values

- In potential flow problems lines of constant value of the stream function are orthogonal to lines of constant value of the potential



Stream Functions (cont.)

- The stream function ψ is related to flux (volume flux is a velocity):

$$F_x = \frac{\partial \psi}{\partial y} \qquad F_y = -\frac{\partial \psi}{\partial x}$$

- If this is substituted into the steady state continuity equation you can see that it is always satisfied

$$\nabla \cdot \mathbf{F} = 0$$

in 2D

$$\frac{\partial F_x}{\partial x} + \frac{\partial F_y}{\partial y} = \frac{\partial^2 \psi}{\partial x \partial y} - \frac{\partial^2 \psi}{\partial x \partial y} = 0$$

Stream Functions (cont.)

- We also know that potential flow implies that there is zero vorticity

$$\nabla \times \mathbf{F} = 0$$

- Substituting the definitions of the fluxes into this equation results in Laplace's equation (strictly speaking the curl is a vector, but in 2D there is only one non-zero component):

$$\nabla^2 \psi = 0$$

Boundary Conditions

- Stream functions are useful if the boundaries are flux boundary conditions

$$F_x = \frac{\partial \psi}{\partial y} \qquad\qquad F_y = -\frac{\partial \psi}{\partial x}$$

- These definitions imply that the stream function is a constant value along a boundary for a zero flux condition and changes linearly for a constant flux
  - Other flux boundaries are readily solvable

Boundary Conditions

There are a few points to note when calculation stream function values around a boundary:

- The velocities in the system depend upon the gradient of the stream function
  - You therefore have 1 degree of freedom when choosing the stream function values
    - Adding a constant to all the stream function values does not change the velocities calculated
  - You can therefore choose any value for the stream function at a single point on the boundary
- Values of the stream function must be continuous around the boundary
  - A discontinuity in the stream function is equivalent to an infinite flux
  - This will let you set the constants of integration obtained when deriving the boundary conditions

An Example

- A metal block 2m wide and 1m high is being heated from below and cooled on the left and right hand sides. The top boundary is thermally insulated
- You can also assume that the block is insulated at the front and back making this a 2D problem
  - It is 1m deep in this direction
- There is a constant heat flux into the bottom of the block of $100W/m^2$
- The heat splits evenly out of left and right hand sides, with a constant flux through each of these sides

What are the values of the stream function around the boundaries of the block?

Calculations

Numerical Solutions to Laplace’s Equation

- Because these are solutions for steady-state systems the solution takes the form of a Boundary Value Problem
  - We need to specify conditions along all the boundaries
  - Need a discretised version of the governing equation to describe all the interior points as a function of their surrounding points

- Non-steady state problems take the form of Initial Value Problems
  - Still need boundary conditions
  - Also require an initial value for all of the locations in the system
    - Hence Initial Value Problem
  - The governing equation needs to not only describe the relationships between neighbouring points in space, but also in time
  - More on these types of problems next week

Finite difference approximation

- Recap from earlier lectures
- Finite difference is the easiest way to approximate differentials on a grid
- There are a number of ways in which first derivatives can be approximated on a grid, 3 of which are:
  - Central difference: $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_{i-1}}{2\Delta x}$ (2nd order accurate)
  - Forward difference: $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_{i+1} - u_i}{\Delta x}$ (1st order accurate)
  - Backward difference: $\left(\frac{\partial u}{\partial x}\right)_i \approx \frac{u_i - u_{i-1}}{\Delta x}$ (1st order accurate)
- 2nd derivatives can also be approximated

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_i \approx \frac{u_{i+1} - 2u_i + u_{i-1}}{\Delta x^2}$$

Finite Difference Approximation of a Linear 2nd Order PDEs

- Using finite difference approximations, any 2D steady state linear 2nd order PDE can be approximated as follows:

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}$$

- The prefactors *a*, *b*, *c*, *d*, *e* and *f* are matrices (i.e. functions of position) in the general case, since the linear coefficients in the PDE can also be functions of position

Approximating Laplaces Equation

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

- Using our earlier approximations for the 2nd derivatives and with $x = x_0 + i\ \Delta x$ and $\mathrm{y} = y_0 + j\ \Delta y$ as the definition for the grid:

$$\nabla^2\psi \approx \frac{\psi_{i+1,j} + \psi_{i-1,j} - 2\psi_{i,j}}{\Delta x^2} + \frac{\psi_{i,j+1} + \psi_{i,j-1} - 2\psi_{i,j}}{\Delta y^2} = 0$$

- If we further assume that the grid is square ($\Delta x = \Delta y$), then we need to solve the following approximation:

$$\psi_{i+1,j} + \psi_{i-1,j} + \psi_{i,j+1} + \psi_{i,j-1} - 4\psi_{i,j} = 0$$

- Using the form on the previous page: $a = b = c = d = 1,\ e = -4, f = 0$

Iterative and Inversion Methods

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}$$

- Since there is one of the above equations for each point in the solution grid, the problem takes the form of a set of coupled linear equations

There are therefore two main approaches that can be used:

- Direct matrix solution or inversion
    - Can reach the solution in one step
    - “Stiff” problems (usually problems with strong convective terms) can result in near singular matrices and thus problems with numerical precision
    - Requires a lot of memory since, while the initial matrix will be sparse, the intermediate matrices in obtaining the solution will not be
- Iterative methods
    - Iteratively approach the solution using a number of steps, each (hopefully) closer to the solution than the previous one
    - Can be easier to implement
        - Improved convergence speed often requires more complexity in the solution
    - Generally has lower memory requirements than inversion techniques

Iterative Methods

- Iterative methods work by successively reducing the error in the solution
- The error we will use is the residual, which is the error in the approximation of the differential equation
  - As we don't know the final solution a priori, we can't use the error in the dependent variable
  - Roughly equivalently we could also use the change in the dependent variable with each iteration as stopping criterion
- The residual at point *i*, *j* is:

$$\xi_{i,j} = a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} - f_{i,j}$$

- …with iterations continuing until the residual drops below a certain value (written here as a relative residual)
  - You might also wish to ensure that the maximum residual is also below a set value

$$\text{Continue iterations until} \frac{\sum\sum|\xi_{i,j}|}{\sum\sum|e_{i,j}u_{i,j}|} < tolerance$$

Jacobi Method

- From the finite difference approximation:

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}$$

- We can rearrange this to set the midpoint value as a function of the neighbouring values:

$$u_{i,j} = \frac{f_{i,j} - \left(a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1}\right)}{e_{i,j}}$$

- We can use this as the basis for an iterative solution:

$$u_{i,j}^{new} = \frac{f_{i,j} - \left(a_{i,j}u_{i+1,j}^{old} + b_{i,j}u_{i-1,j}^{old} + c_{i,j}u_{i,j+1}^{old} + d_{i,j}u_{i,j-1}^{old}\right)}{e_{i,j}}$$

Implement in Python

- I will now implement Jacobi method in Python
  - Very simple algorithm to implement
    - Not just for potential flow problems. It can be used for solving any linear PDE
  - Good for testing simple problems on square grids

- Later in this course you will be learn about solving PDEs using the Finite Element method
  - Easier to apply to more complex boundary shapes
  - …, but a lot more complex to implement

Steps in solving this problem

- Derive boundary conditions/values for each location on the boundary
  - We will only be using Dirichlet boundary conditions in this example
  - We therefore need the value of stream function at each grid point on the edge of the grid
- Discretise the governing equation
  - We have already done this for Laplace's equation
- Give an initial guess for the stream function value at every point inside the grid
  - A good guess can be critical for the convergence of some systems, but it is not very important here as solutions of Laplace's equation with Dirichlet boundary conditions converges quickly
- Implement the iterative solver
  - Calculate a new guess for the grid values based on the previous guess
  - Repeat until the residual (or change in the dependent variable) is sufficiently small

Systems with more complex Boundaries

- Thus far we have only considered problems with Dirichlet boundary conditions
  - Values of the dependent variable specified along the boundary
- These are easier to solve numerically (and have better convergence behaviour) than Neumann and other more complex boundary conditions
  - It also makes analytical solutions tractable
- When solving numerically using SOR, the more complex boundary conditions will require that the points on the boundary are included in the iteration

Neumann Boundary Conditions

- This type of boundary condition is when the gradient of the dependent variable is specified at the boundary:

$$\nabla u \cdot \hat{n} = g(x, y)$$

- Where $\hat{n}$ is the unit normal to the boundary

- This means that:
  - For a vertical boundary: $$\frac{\partial u}{\partial x} = g(y)$$
  - ...and for a horizontal boundary: $$\frac{\partial u}{\partial y} = g(x)$$

Neumann Boundary Conditions (cont.)

- This boundary condition is often found in potential flow problems
  - E.g. If the flux through the boundary is specified and the potential is the dependent variable

$$F_{In} = -k\nabla\varphi \cdot \hat{\mathbf{n}}$$

- Where F_{In} is the flux in through the boundary and the unit normal points into the domain

- A no flux boundary condition in potential flow is thus a zero gradient, which is a special case of the Neumann boundary condition

Finite Difference Approximation of a Neumann Boundary

- As the point being considered is, by definition, on the edge of the valid domain, the central difference can't be used.
- Whether a forward or backward approximation is used depends on which side of the domain the boundary is
  - E.g. If forward difference in the *x* direction on the left hand boundary and backward difference on the right hand boundary
  - Similarly, in the *y* direction a forward difference needs to be used on the bottom boundary and a backward difference on the top boundary

Finite Difference Approximation of a Neumann Boundary (cont.)

- If the finite difference approximation is expressed as follows:

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}$$

- ... and x_0, y_0 is the bottom left hand corner then:

Vertical Boundaries: $\frac{\partial u}{\partial x} = g(y)$

- For the left hand boundary:

  $a_{i,j}=1$ $b_{i,j}=c_{i,j}=d_{i,j}=0$ $e_{i,j}=-1$ $f_{i,j}=\Delta x\, g(y_j)$

- For the right hand boundary:

  $b_{i,j}=-1$ $a_{i,j}=c_{i,j}=d_{i,j}=0$ $e_{i,j}=1$ $f_{i,j}=\Delta x\, g(y_j)$

Horizontal Boundaries: $\frac{\partial u}{\partial y} = g(x)$

- For the bottom boundary:

  $c_{i,j}=1$ $a_{i,j}=b_{i,j}=d_{i,j}=0$ $e_{i,j}=-1$ $f_{i,j}=\Delta y\, g(x_i)$

- For the top boundary:

  $d_{i,j}=-1$ $a_{i,j}=b_{i,j}=c_{i,j}=0$ $e_{i,j}=1$ $f_{i,j}=\Delta y\, g(x_i)$

Robin Boundary Condition

- The third type of linear PDE boundary condition is the Robin boundary condition
- This condition is a linear combination of the Dirichlet and Neumann boundary conditions:

$$\nabla u \cdot \hat{n} + g(x, y)u = h(x, y)$$

This means that:

- For a vertical boundary: $$\frac{\partial u}{\partial x} + g(y)u = h(y)$$
- ...and for a horizontal boundary: $$\frac{\partial u}{\partial y} + g(x)u = h(x)$$

Finite Difference Approximation of a Robin Boundary (cont.)

- Again using the following form for the approximation:

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}$$

Vertical Boundaries:

$$\frac{\partial u}{\partial x} + g(y)u = h(y)$$

- For the left hand boundary:

  $a_{i,j}=1$ $\quad b_{i,j}=c_{i,j}=d_{i,j}=0$ $\quad e_{i,j}=-1 + \Delta x\, g(y_j)$ $\quad f_{i,j}=\Delta x\, h(y_j)$

- For the right hand boundary:

  $b_{i,j}=-1$ $\quad a_{i,j}=c_{i,j}=d_{i,j}=0$ $\quad e_{i,j}=1 + \Delta x\, g(y_j)$ $\quad f_{i,j}=\Delta x\, h(y_j)$

Horizontal Boundaries:

$$\frac{\partial u}{\partial y} + g(x)u = h(x)$$

- For the bottom boundary:

  $c_{i,j}=1$ $\quad a_{i,j}=b_{i,j}=d_{i,j}=0$ $\quad e_{i,j}=-1 + \Delta y\, g(x_j)$ $\quad f_{i,j}=\Delta y\, h(x_i)$

- For the top boundary:

  $d_{i,j}=-1$ $\quad a_{i,j}=b_{i,j}=c_{i,j}=0$ $\quad e_{i,j}=1 + \Delta y\, g(x_j)$ $\quad f_{i,j}=\Delta y\, h(x_i)$

Where do Robin Boundary Conditions arrise

- Robin boundary conditions are encountered in convective-diffusive problems
  - Convective-diffusive problems are ones in which a flux is proportional to the sum of a zero order (convective) term and a 1^{st} order (diffusive) term
    - Note that when substituted into a continuity equation the convective term becomes 1^{st} order and the diffusive 2^{nd} order
  - A flux with a convective and diffusive component takes the following form:

$$\mathbf{F} = \mathbf{v}u - D\nabla u$$

- Where $\mathbf{F}$ is the flux of u. $\mathbf{v}$ is the convective velocity vector and D is the diffusion coefficient

Robin Boundary Condition (cont.)

- If the flux through the boundary is F_{in} and $\hat{\boldsymbol{n}}$ is the unit normal to the boundary pointing into the domain, then the flux boundary condition for a convective-diffusive problem is:

$$F_{in} = (\mathbf{v} \cdot \hat{\mathbf{n}})u - D\nabla u \cdot \hat{\mathbf{n}}$$

- This thus takes the form of a Robin boundary condition

Note on Dirichlet boundary

- If mixed boundary conditions are being used it is often easier to simply include any Dirichlet boundaries in the iterations as well
  - Potentially easier than having special treatment for these boundaries

- If the finite difference approximation is expressed as follows:

$$a_{i,j}u_{i+1,j} + b_{i,j}u_{i-1,j} + c_{i,j}u_{i,j+1} + d_{i,j}u_{i,j-1} + e_{i,j}u_{i,j} = f_{i,j}$$

- The Dirichlet condition is that the value is a function of position on the boundary

$$u = g(x, y)$$

- The parameters in the above form are thus as follows

$$a_{i,j} = b_{i,j} = c_{i,j} = d_{i,j} = 0 \qquad e_{i,j} = 1 \quad f_{i,j} = g(x_i, y_j)$$