Chapter 14
Potential and Flow Visualization

14.1 Definition and First Examples

Potential and streamfunction are mathematical functions that cannot be observed directly in the real world, but which turn out to be extremely powerful concerning the calculation and visualization of 2D flow fields. There are applications for all types of fluids, for free flow of gases and liquids, as well as for porous media flow. Electro- and magnetodynamics are other scientific fields where potential theory is applied extensively.

The notation potential refers to a function $\varphi$, from which a flow field is derived by the gradient of $\varphi$. $\varphi$ is a velocity potential if:

$$v = \pm \nabla \varphi$$

(14.1)

For steady incompressible fluids (see Chap. 2), for which the continuity equation $\nabla \cdot v = 0$ is valid, follows the potential equation or Laplace\(^1\) equation\(^2\):

$$\nabla^2 \varphi = 0, \quad \text{in 2D: } \frac{\partial^2 \varphi(x, y)}{\partial x^2} + \frac{\partial^2 \varphi(x, y)}{\partial y^2} = 0$$

in 3D:

$$\begin{align*}
\frac{\partial^2 \varphi(x, y, z)}{\partial x^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial y^2} + \frac{\partial^2 \varphi(x, y, z)}{\partial z^2} &= 0 \\
(14.2)
\end{align*}$$

The short form, using the $\nabla$-operator, is valid for 2D and 3D cases. In fluid dynamics the potential $\varphi$ has the physical unit of $[\text{m}^3/\text{s}]$. The name potential is connected with the property that at each location of the model region the flux or velocity vector can be derived from the gradient of the potential:

---

\(^1\) Pierre-Simon Laplace (1749–1827), French mathematician and astronomer.

\(^2\) The formulation $\Delta \varphi = 0$ can be found frequently, which makes sense, as the Laplace operator $\Delta$ is formally defined as $\Delta := \nabla \cdot \nabla$. 
\[ v = \nabla \varphi \quad \text{(14.3)} \]

There is an entire mathematical discipline dealing with solutions of the 2D potential equation. Subject of complex analysis are harmonic functions that are solutions of the 2D Laplace equation. In this chapter we deal with 1D parallel flow that is represented by the potential

\[ \varphi(x, y) = \varphi_0 + \varphi_x x + \varphi_y y \quad \text{(14.4)} \]

Additionally there are sources and sinks in the infinitely extended space, which are represented by the potential:

\[ \varphi(x, y) = \frac{Q}{2\pi} \log(|r - r_0|) \quad \text{(14.5)} \]

where \(\varphi_x, \varphi_y\) and \(\varphi_0\) are constant numbers. \(Q\) denotes the source- or sink-rate, \(r_0\) the location of the source or sink in 2D space and \(r = (x, y)\) the vector towards the current location. According to vector analysis, \(r - r_0\) is the vector connecting source/sink location with the current position. \(|r - r_0|\) is the length of the connecting vector, equal to \(\sqrt{(x-x_0)^2 + (y-y_0)^2}\).

In the following we examine potentials emerging from the superposition of formulae (14.4) and (14.5). According to the principle of superposition, the sum of the functions is a solution of the Laplace equation too. The principle is a trivial consequence from the fact that the potential equation is linear. In MATLAB® such functions can be visualized easily as demonstrated by the following command sequence.

```matlab
xmin = -1; xmax = 1;                       % x-coordinates
ymin = 0; ymax = 2;                        % y-coordinates
x0 = 0; y0 = .905;                         % source/sink location
Qx0 = 0.1; Qy0 = 0;                        % baseflow components
Q = 1;                                     % source/sink rate

% mesh generation
xvec = linspace(xmin,xmax,100);           % x-coordinates
yvec = linspace(ymin,ymax,100);           % y-coordinates
[x,y] = meshgrid(xvec,yvec);              % create mesh

% processing
r = sqrt((x-x0).^2+(y-y0).^2);             % distances to well
phi = -Qx0*x - Qy0*y + (Q/(2*pi))*log(r);  % potential

%post-processing
surf (x,y,phi);                           % surface plot
```

With the first five instruction lines the parameter values are specified. In the next step, 100 equidistant positions at the intervals on the \(x\)- and \(y\)-axis are computed before the mesh is constructed and stored in the \(x\) and \(y\) arrays, using MATLAB® meshgrid. Next, the array \(r\) is computed, which contains the distances to the