OSCAR CORTEZ

SYMBOLIC MATHEMATICS WITH SYMPY

@OSKRGAB

SymPy Example Using the Diffusivity Equation

The radial form of the diffusivity equation for slightly compressible fluids can be written in the following form:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} = -\frac{1}{\eta}\frac{\partial p}{\partial t}$$

Recall the assumptions and limitations of this form of the diffusivity equation:

1. Homogeneous and isotropic porous medium
2. Uniform thickness
3. Single-phase flow
4. Laminar flow
5. Rock and fluid properties independent of pressure

To show a solution to the diffusivity equation, we will use a steady-state flow condition, i.e., $\frac{\partial p}{\partial t} = 0$, and therefore the diffusivity equation reduces to:

$$\frac{\partial^2 p}{\partial r^2} + \frac{1}{r}\frac{\partial p}{\partial r} = 0$$

The previous equation is called Laplace's equation for steady-state flow. Which we will solve using SymPy.

1. Define the function to be solved

In this case we want to solve for the pressure function, $p(r)$, which is a function of the radial distance, r. We will use the `Function` class from SymPy to define the function.

In [1]:

```
from sympy import Function, symbols, Eq, dsolve
from sympy.abc import r

# Define the pressure function
p = Function('p')(r)
p
```

Out[1]:

$p(r)$

2. Define the differential equation

Let's define the Laplace equation for steady-state flow in SymPy. We will use the `Eq` class to define the equation.

In [2]:

```
# Define the differential equation as shown in the image
laplace_eq = Eq(p.diff(r, r) + (1/r) * p.diff(r), 0)
laplace_eq
```

Out[2]:

$$\frac{d^2}{dr^2}p(r) + \frac{\frac{d}{dr}p(r)}{r} = 0$$

3. Define the initial conditions

We will define the initial condition as $p(r_w) = p_{wf}$, where r_w is the wellbore radius and p_{wf} is the wellbore flowing pressure. We will use the `subs` method to substitute the initial condition into the differential equation.

In [3]:

```
# Define symbols for the initial conditions
pwf, rw = symbols('pwf rw')

# Define initial condition P(rw) = pwf
ics={p.subs(r, rw): pwf}
```

4. Solve the differential equation

We will use the `dsolve` function to solve the differential equation. We will pass the differential equation and the initial conditions as arguments to the `dsolve` function.

In [4]:

```
# Solve the differential equation
solution = dsolve(laplace_eq, p, ics=ics)

# Display the solution
solution
```

Out[4]:

$$p(r) = C_2 \log(r) - C_2 \log(rw) + pwf$$

5. Arrange the equation.

We can manually rearrange the solution to get the radial form of the Darcy equation as follows:

$$p(r) = p_{wf} + C_1\, ln(r/r_w)$$

Where $C_1 = \frac{Q_o B_o \mu_o}{0.00708kh}$