PT2 - Hands On #1 - SED3

Felix Lechleitner

Introduction

This is the report document for the solution of the SED3 problem. The solution is coded in R and the report is written in RMarkdown and compiled to a pdf-document using knitr + LaTeXand to a html-webpage using knitr. The source code can be found here: Click me!

The report as a pdf-file can be found here: PDF-Report (Hyperlink)

The report as webpage can be found here (code might be easier to read as html): HTML-Report (Hyperlink)

Results from code block will be displayed at the end of the respective block.

Given values

The given values for the problem are derived from the authors name and are as follows:

Table 1: Name-specific variables.

	Value	Unit
Temperature	22	°C
Shape	Tetrahedron	-
Sedimenation velocity	0.005	m s^{-1}
Mass loading	0.14	-

Unit conversions and temperature dependent functions

Before we turn to the task at hand, we define the needed variables and functions. The value for the viscosity of water is missing in the problem specification and has been calculated for the given temperature and at ambient pressure using this online-tool: http://www.peacesoftware.de/einigewerte/wasser_dampf.html

```
# clear workspace ------------------------------------------------------------
rm(list = ls())

# install pacman for easy installation/loading of libraries ----------------
if (!require("pacman")) {
  install.packages("pacman")
}


# load libraries -------------------------------------------------------------
pacman::p_load(dplyr)


# assumptions ----------------------------------------------------------------
pressure <- 101325 # [Pa]

# given values ---------------------------------------------------------------
temperature <- 22 # [°C]
u.sed <- 5 * 10 ^ -3 # [m/s]
```

```r
mass.loading <- 0.14 # [-]
diameter.silica <- 1.2 # [µm]
density.particle <- 2320 # [kg/m^3]
dyn.viscosity.fluid <- 0.00095442328933174 # [Pa s]


# unit conversions ------------------------------------------------------
temperature <- temperature + 273.15 # [K]
diameter.silica <- diameter.silica * 10 ^ -6 # [m]



# defince functions for silica density and fluid density ----------------
CalculateSilicaDensity <- function(temperature) {
  return(2440 - 5.44 * 10 ^ -2 * temperature)
}

CalculateFluidDensity <- function(temperature) {
  return((
    1.024 - 0.0002575 * (temperature - 273.15) - 2.793 * 10 ^ -6 * (temperature -
                                                                      273.15) ^ 2
  ) * 1000)
}

# calculate temp-dependent values ---------------------------------------

density.silica <- CalculateSilicaDensity(temperature)
density.fluid <- CalculateFluidDensity(temperature)
```

$\rho_{\text{silica}} = 2423.94384 \text{ kg m}^{-3}$

$\rho_{\text{fluid}} = 1016.983188 \text{ kg m}^{-3}$

$T = 295.15 \text{ K}$

$d_{\text{silica}} = 1.2\text{e-}06 \text{ m}$

Task 1

Calculating the properties of the suspension

The first thing that will be calculated are the properties of the suspension/mixture. The mass fraction of silica can be calculated from the mass loading. The mass loading can be used to calculate the the density of the mixture. From the density of the mixture, the silica and the fluid the volume fraction of silica and the viscosity of the mixture can be calculated.

```r
# calculate mixture properties ------------------------------------------

massfrac.silica <- mass.loading / (1 + mass.loading)

density.mixture <-
  1 / ((massfrac.silica / density.silica) + ((1 - massfrac.silica) / density.fluid))

volfrac.silica <-
  (density.mixture - density.fluid) / (density.silica - density.fluid)
```

```
dyn.viscosity.mixture <- dyn.viscosity.fluid * (1 + 2.5 * volfrac.silica)
```

$w_{\text{silica}} = 0.12280701754386$

$\rho_{\text{mixture}} = 1095.04033990573 \text{ kg m}^{-3}$

$\phi_{\text{silica}} = 0.0554792714314851$

$\eta_{\text{mixture}} = 0.00108680006115516 \text{ Pa s}$

The next step is to remove the influence of the particle shape from the measured sedimentation velocity.

```
# size correction for measured sedimentation velocity ----------------------

shape.factor <- 0.68
u.sed.sphere <- u.sed / shape.factor
```

$\phi_{\text{shape}} = 0.68$

$u_{\text{Sphere}} = 0.00735294117647059 \text{ m s}^{-1}$

Using the sedimentation velocity of the equivalent sphere, we can calculate the Ljaščenko number.

```
# calculate Ljaščenko number from corrected u_sed --------------------------

Lj <-
  (u.sed.sphere ^ 3 * density.mixture ^ 2) /
  (dyn.viscosity.mixture * 9.81 * (density.particle - density.mixture))
```

$\text{L}_\text{j} = 0.0365008571463289$

Based on L_j we can calculate the Archimedes number and using the definition of said number the particle diameter of the equivalent sphere.

```
# calculate Ar based on value of Lj -----------------------------------

Ar <- case_when(
  Lj >= 0 & Lj <= 0.014 ~ (Lj / (1.72 * 10 ^ -4)) ^ (1 / 2),
  Lj > 0.014 & Lj <= 3.12 ~ (Lj / (5.41 * 10 ^ -4)) ^ (1 / 1.5),
  Lj > 3.12 & Lj <= 167 ~ (Lj / (4.28 * 10 ^ -3)) ^ (1 / 1.143),
  Lj > 167 & Lj <= 3200 ~ (Lj / (5.1 * 10 ^ -2)) ^ (1 / 0.875),
  Lj > 3200 & Lj <= 6.1 * 10 ^ 5 ~ (Lj / 5.4) ^ (1 / 0.5)
)


# calculate particle diameter from Ar ---------------------------------

diameter.sphere <-
  ((Ar * dyn.viscosity.mixture ^ 2) / (9.81 * (density.particle - density.mixture) *
                                         density.mixture)) ^ (1 / 3)
```

$\text{Ar} = 16.5731043682141$

$d_{\text{Sphere}} = 1.141548\text{e-}04 \text{ m}$

The volume of the sphere can now be calculated from the diameter. The volume of the platonic solid needs to have the same volume as the sphere, so we can calculate the edge length of the Tetrahedron by equating the volumina of the two bodies.

```
# calculate sphere equivalent volume and edge length ----------------------

sphere.volume <- diameter.sphere ^ 3 * pi / 6

edge.length <- (sphere.volume * 6 * sqrt(2)) ^ (1 / 3)
```

$V_{\text{Sphere}} = V_{\text{Tetrahedron}} =$ 7.78898528490098e-13 m^3

$a_{\text{Tetrahedron}} =$ 1.876646e-04 m

The Reynolds number can be calculated using the diameter of the equivalent sphere or the edge length of the tetrahedron and the actual sedimentation velocity. Furthermore, all values that are needed for the calculation of the forces acting on the particle are known.

```
Re.tetrahedron <-
  edge.length * u.sed * density.particle / dyn.viscosity.mixture

Re.sphere <-
  diameter.sphere * u.sed * density.particle / dyn.viscosity.mixture

force.gravity <- sphere.volume * density.particle * 9.81
force.buoyancy <- sphere.volume * density.mixture * 9.81
force.drag <- force.gravity - force.buoyancy
```

$\text{Re}_{\text{Tetrahedron}} =$ 2.00304502795656

$\text{Re}_{\text{Sphere}} =$ 1.21843515790259

$F_{\text{Gravity}} =$ 1.77271073896118e-08 N

$F_{\text{Buoyancy}} =$ 8.3671972851146e-09 N

$F_{\text{Drag}} =$ 9.35991010449724e-09 N

Task 1 is complete.

Task 2

For task 2, the forces acting on the sphericle particle are the same as in task 1, as gravity and buoyancy are dependent on the volume which does not change. If we assume creeping flow, Re needs to be below 0.5. For creeping flow we can calculate the sedimentation velocity like this:

$$u_{\text{sed, St}} = \frac{d_{\text{p}} \cdot (\rho_{\text{p}} - \rho_{\text{mixture}}) \cdot g}{18\,\eta}$$

```
u.sed.stokes <-
  diameter.sphere ^ 2 *
  (density.particle - density.mixture) * 9.81 / (18 * dyn.viscosity.mixture)

Re.stokes <-
  diameter.sphere * u.sed.stokes * density.particle / dyn.viscosity.mixture
```

$u_{\text{Stokes}} = 0.00800491655397925 \text{ m s}^{-1}$

$\text{Re}_{\text{Stokes}} = 1.95069435308896$

However, the Reynolds number we get from this calculation does not indicate creeping flow, so the equation we used can't be applied in this case.

We could solve this problem iteratively. We take the Reynolds number from the previous task as starting value, the calculate the correct drag coefficient and the sedimentation velocity from the drag force. with this new velocity we can recalculate the Reynolds number and repeat this process until the change of the Reynolds number between iteration steps is neglible.

```
Re.stokes <- Re.sphere

area.projection.sphere <- diameter.sphere ^ 2 * pi / 4
iteration.counter <- 0

while (TRUE) {
  drag.coeff <- case_when(
    Re.stokes <= 0.5 ~ 24 / Re.stokes,
    Re.stokes > 0.5 & Re.stokes <= 10 ~ 27 / Re.stokes ^ 0.8,
    Re.stokes > 10 & Re.stokes <= 120 ~ 17 / Re.stokes ^ 0.6,
    Re.stokes > 120 & Re.stokes <= 1000 ~ 6.5 / Re.stokes ^ 0.4,
    Re.stokes > 1000 & Re.stokes <= 2 * 10 ^ 5 ~ 0.44

  )

  u.sed.stokes <-
    (2 * force.drag / (drag.coeff * area.projection.sphere * density.mixture)) ^
    0.5

  Re.stokes.previous.it <- Re.stokes
  Re.stokes <-
    diameter.sphere * u.sed.stokes * density.mixture / dyn.viscosity.mixture

  iteration.counter <- iteration.counter + 1

  if (abs(Re.stokes.previous.it - Re.stokes) < 10 ^ -20) {
    break
  }

}
```

Calculation converged after 41 iterations.

$u_{\text{Stokes}} = 0.00735711955743093 \text{ m s}^{-1}$

$\text{Re}_{\text{Stokes}} = 0.846218206110681$

This is however not the Stokes sedimentation velocity as we are actually already in the transition regime. The calculated velocity is the same as the shape corrected sedimentation velocity.