

1. Bodensaugspannung in nutzbare Feldkapazität (nFK) umwandeln

Samantha Rubo

2021-12-08

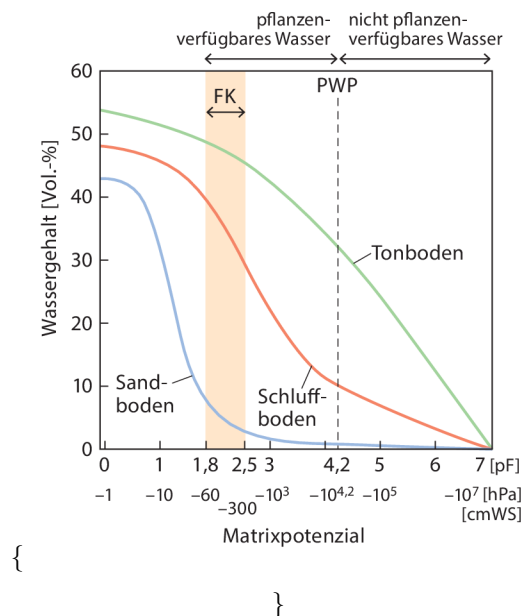
Wasserhaltevermögen der Felder der HGU:

Schicht Nr.	Bezeichnung	100%nFK entsprechen	“Wassergehalt (Vol.% bei 100%nFK)”
1	0-30 cm	49.3 mm	16.4%
2	30-60 cm	46.1 mm	15.4%
3	60-90 cm	43.4 mm	14.5%

Daten unter “Z:\Außenbetrieb\Flächenbelegung\Wasserhaltevermögen Böden Felder Gb.xlsx”

```
knitr::include_graphics("../graphics/Feldkapazität_Thomas.png")
```

```
\begin{figure}
```



\caption{Beziehung Volumetrischer Wassergehalt ~ Matrixpotential. Berechnungsgrundlage der nFK. Quelle: Thomas F. (2018) Ökophysiologische Leistungen der Höheren Pflanzen. In: Grundzüge der Pflanzenökologie. Springer Spektrum, Berlin, Heidelberg.

https://doi.org/10.1007/978-3-662-54139-5_3%7D/label%7Bfig:unnamed-chunk-1%7D \end{figure}

Grenzwerte zur Berechnung der nutzbaren Feldkapazität

```
pf_min = 1.8  
pf_max = 4.2
```

```

hPa_min = 10^pf_min
hPa_max = 10^pf_max
#Vol% Wasser bei 100% nFK für drei Bodentiefen:
T0020 <- 49.3
T4060 <- 46.1
T2040 <- T0020 - (T0020-T4060)/2

```

Tensiometer-Daten, Wetter und Bewässerung aus SQLite-Datenbank lesen.

```

#Verbindung zur Datenbanl herstellen:
path1 <- ifelse(Sys.info()["nodename"] == "Samanthas-MBP.local",
               "../..../Github_GeoSenSys/GeoSenSys2020/", "../..../GeoSenSys2020/")
db <- paste0(path1, "Data_2020/HGU_GeoSenSys_V3_6.db") #DB in other R-Project

db1 <- dbConnect(RSQLite::SQLite(), db)

#Query fuer Tensiometer-Datensatz
query <- "SELECT
  Spinat_Saetze.satz_id,
  Varianten.variante_H2O,
  Parzellen.parzelle_name,
  Tensiometer.zeit_messung,
  Tensiometer.bodensaugspannung_0020_hPa,
  Tensiometer.bodensaugspannung_2040_hPa,
  Tensiometer.bodensaugspannung_4060_hPa

  FROM Tensiometer
  LEFT JOIN Parzellen ON Tensiometer.parzelle_id = Parzellen.parzelle_id
  LEFT JOIN Varianten ON Parzellen.variante_id = Varianten.variante_id
  LEFT JOIN Spinat_Saetze ON Varianten.satz_id = Spinat_Saetze.satz_id
"

tensio <- dbGetQuery(db1, query)

#### Query fuer Wetter-Daten:
query2 <- "SELECT
  Wetter.datum_wetter,
  Wetter.niederschlag_mm
  FROM
  Wetter"

wetter <- dbGetQuery(db1, query2)%>% #Niederschlag aller Saetze einlesen
mutate_at("datum_wetter", ~as_date(.))

#Query fuer Bewaesserungs-Datensatz
query3 <- "SELECT
  Spinat_Saetze.satz_id,
  Varianten.variante_H2O,
  Parzellen.parzelle_name,
  Bewaesserung.datum_bewaesserung,
  Bewaesserung.wassermengen_mm

```

```

FROM Bewaesserung
LEFT JOIN Parzellen ON Bewaesserung.parzelle_id = Parzellen.parzelle_id
LEFT JOIN Varianten ON Parzellen.variante_id = Varianten.variante_id
LEFT JOIN Spinat_Saetze ON Varianten.satz_id = Spinat_Saetze.satz_id
"
bewaesserung <- dbGetQuery(db1, query3) %>% #Bewaesserung aller Saetze einlesen
mutate_at("datum_bewaesserung", ~as_date(.))

dbDisconnect(db1) #Verbindung zur Datenbank beenden
rm(db, db1, path1, query, query2, query3) #Helfer-Objekte loeschen

```

Daten formatieren und Tagesmittelwerte der Bodensaugspannung berechnen

```

tensio <- tensio %>%
  #Datum formatieren und Tagesmittelwerte bilden
  mutate_at("zeit_messung", ~as_datetime()) %>%
  mutate_at("zeit_messung", ~format.Date(., format="%Y-%m-%d")) %>% #für Tageweise Mittelwert
  mutate_at("zeit_messung", ~as_date()) %>% #wieder in Datum (class) umformen

  #Varianten nach Bewässerungs-Stufe aufteilen:
  rename(wasser_level = variante_H2O) %>%
  #Faktorstufen sortieren (für Grafik)
  mutate_at("wasser_level", ~factor(., levels = c("Wfull_plus", "Wfull", "Wred"))) %>%
  group_by(satz_id, wasser_level, zeit_messung) %>%
  summarise_at(c("bodensaugspannung_0020_hPa",
                 "bodensaugspannung_2040_hPa",
                 "bodensaugspannung_4060_hPa"), ~round(mean(., na.rm=TRUE) , digits = 2))

```

hPa in pf umwandeln: log10(hPa)

```

tensio <- tensio %>%
  mutate(pf_0020 = log10(bodensaugspannung_0020_hPa),
         pf_2040 = log10(bodensaugspannung_2040_hPa),
         pf_4060 = log10(bodensaugspannung_4060_hPa)
  )

```

nFK berechnen

```

nfk_fun <- function(x){(1-(x-pf_min) / (pf_max-pf_min))*100}
tensio <- tensio %>% #as_tibble() %>%
  mutate_at(c("pf_0020", "pf_2040", "pf_4060"),
            list(nfk = nfk_fun)) %>%
  rename_with(.cols = ends_with("_nfk"), ~gsub("pf_", "T", .x, fixed = TRUE)) %>%
  # Tabelle komprimieren: 2 Nachkommastellen
  mutate_if(is.numeric, ~round(., 2))

## `mutate_if()` ignored the following grouping variables:
## Columns `satz_id`, `wasser_level`

```

Tabelle formatieren für ggplot

```
tensio_melted <- tidyr::pivot_longer(tensio %>%
  select(c("satz_id", "wasser_level", "zeit_messung",
           "T0020_nFK", "T2040_nFK", "T4060_nFK")),
  cols = c("T0020_nFK", "T2040_nFK", "T4060_nFK"),
  names_to = "Bodentiefe",
  values_to = "nFK_prozent") %>%
mutate(Bodentiefe = substr(Bodentiefe, 4, 5) %>% as.numeric) %>%
mutate(kategorie = factor("nFK", levels=c("wasserinput", "nFK")))
```

#Tensio bei 0cm erweitern

```
tensio0 <- tensio_melted %>% filter(Bodentiefe == 20) %>%
  mutate(Bodentiefe = 0)

tensio_melted <- bind_rows(tensio_melted, tensio0)

rm(tensio0)
```

Wetter- und Bewässerungsdaten formatieren und zusammenführen

```
bewaesserung_mean <- bewaesserung %>%
  rename(wasser_level = variante_H2O) %>%
  group_by(satz_id, datum_bewaesserung, wasser_level) %>%
  summarise(bewaesserung_mm = mean(wassermengen_mm)) %>%
  ungroup
```

`summarise()` has grouped output by 'satz_id', 'datum_bewaesserung'. You can override using the `.groups` argument.

#Bewaesserung und Niederschlag in eine Tabelle zusammenfuehren

```
wasser_gesamt <- tensio %>% select(satz_id, wasser_level, zeit_messung) %>%
  left_join(wetter, by=c("zeit_messung" = "datum_wetter")) %>%
  left_join(bewaesserung_mean, by = c("satz_id", "wasser_level", "zeit_messung" = "datum_bewaesserung")) %>%
  #Faktorstufen sortieren (für Grafik)
  tidyr::pivot_longer(cols = c("bewaesserung_mm", "niederschlag_mm"),
    names_to = "variable", values_to = "value") %>%
  mutate(kategorie = factor("wasserinput", levels=c("wasserinput", "nFK"))) %>%
  mutate_at("wasser_level", ~factor(., levels = c("Wfull_plus", "Wfull", "Wred")))
```

nFK plotten

plot erstellen: #minus-Bodentiefe, da der Wert die "bis"-Bodentiefe beschreibt

```
plot_nfk <- function(satz_nr, subtitle){
  my_breaks <- c(seq(from = 0, to = 120, by = 10), Inf)
  my_colors <- c(colorRampPalette(c("brown", "#fac83c"))(6),
    colorRampPalette(c("lightgreen", "forestgreen"))(3),
    colorRampPalette(c("#32c8fa", "darkblue"))(4))
  my_labels <- levels(cut(tensio_melted$nFK_prozent, breaks = my_breaks))

  p <- ggplot() +
    geom_contour_filled(data = tensio_melted %>% filter(satz_id == satz_nr),
      aes(x = zeit_messung, y = -(Bodentiefe), z = nFK_prozent),
      breaks = my_breaks) +
    scale_colour_manual("Soil moisture (% nFK)",
```

```

        values = my_colors, labels = my_labels,
        aesthetics = "fill", drop = FALSE
    ) +
    # "drop = FALSE" ist notwendig um auch Werte anzuzeigen (und die Farben zu scalen),
    # die nicht ueber den Wertebereich in der CSV hinausgehen (z.B. <10 oder >120 % nFK)
    labs(title = "Bodenfeuchte und Wassereintrag für drei Behandlungen",
         subtitle = subtitle,
         x = "Date",
         y = "\"#\"Soil depth (cm)\"")
    ) +
    ggnewscale::new_scale("fill") + #zweite fill-scale für Regen und Bewässerung
    # Regen und Bewässerung anfügen
    geom_col(data = wasser_gesamt %>% filter(satz_id == satz_nr),
            aes(zeit_messung, value, fill = variable)) +
    scale_fill_manual("Wassereintrag (mm)",
                     values = c("niederschlag_mm" = "gray70",
                                "bewaesserung_mm" = "gray40"),
                     labels = c("niederschlag_mm" = "Regen",
                                "bewaesserung_mm" = "GS")) +
    facet_grid(kategorie~wasser_level, scales = "free_y", switch = "y", space = "free_y",
              labeller = labeller(.rows = as_labeller(c(nFK = "Bodentiefe (cm)",
                                                         wasserinput = "Wassereintrag (mm)")),
                                .cols = label_value),
              drop = FALSE) + #damit auch Wfull_plus in Satz1 dargestellt wird (ohne Daten)
    theme_bw() + theme(panel.grid = element_blank(),
                      strip.placement = "outside",
                      strip.background.y = element_blank(),
                      panel.spacing = unit(0, "lines")
                      #strip.switch.pad.grid = unit('0.0', "cm"))
  )
  # Ausgabe des Plots
  p
}

```

#Plot-Funktion ausfuehren

```

p2 <- plot_nfk(satz_nr = 2, subtitle = "2020, Satz 2, Feld 4a (*Paper VI über diese Daten)")
p3 <- plot_nfk(satz_nr = 3, subtitle = "2021, Satz 1, Feld 6")
p4 <- plot_nfk(satz_nr = 4, subtitle = "2021, Satz 2, Feld 4b")
p5 <- plot_nfk(satz_nr = 5, subtitle = "2021, Satz 3, Feld 6")

plot_list <- list(p2, p3, p4, p5)
#p2; p3; p4; p5

```

#Grafik speichern

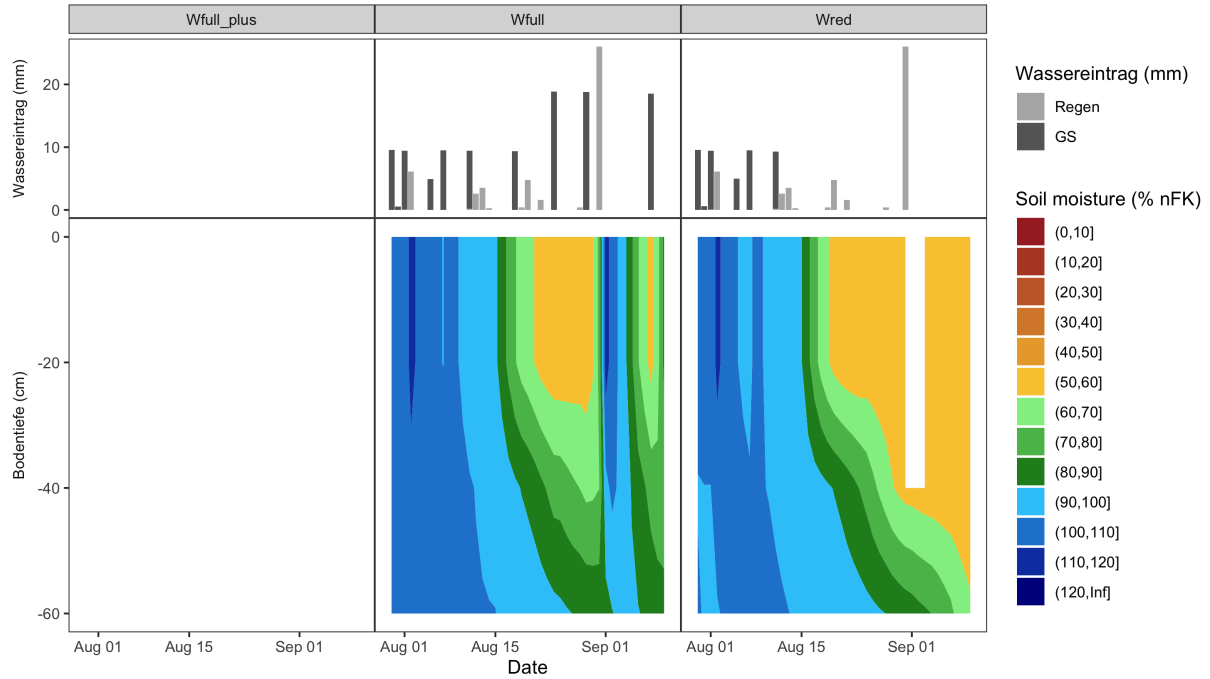
```

# file_list <- list(
# #file1 <- #keine Tensiometer-Daten fuer Satz 1
# file2 = "../graphics/nFK_2020_Satz2.png",
# file3 = "../graphics/nFK_2021_Satz1.png",
# file4 = "../graphics/nFK_2021_Satz2.png",
# file5 = "../graphics/nFK_2021_Satz3.png")
#
# purrr::map2(file_list, plot_list,
#             ~ggsave(filename = .x, plot = .y, device = "png", width = 10, height = 6, dpi = 300))

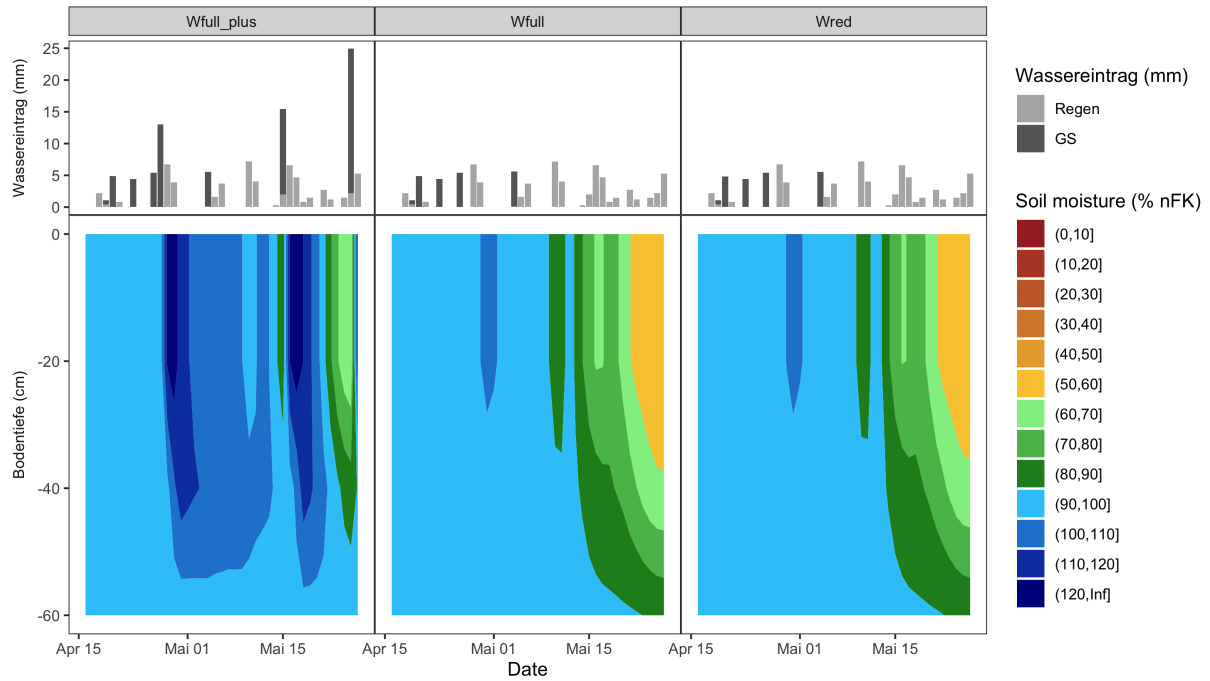
```

)

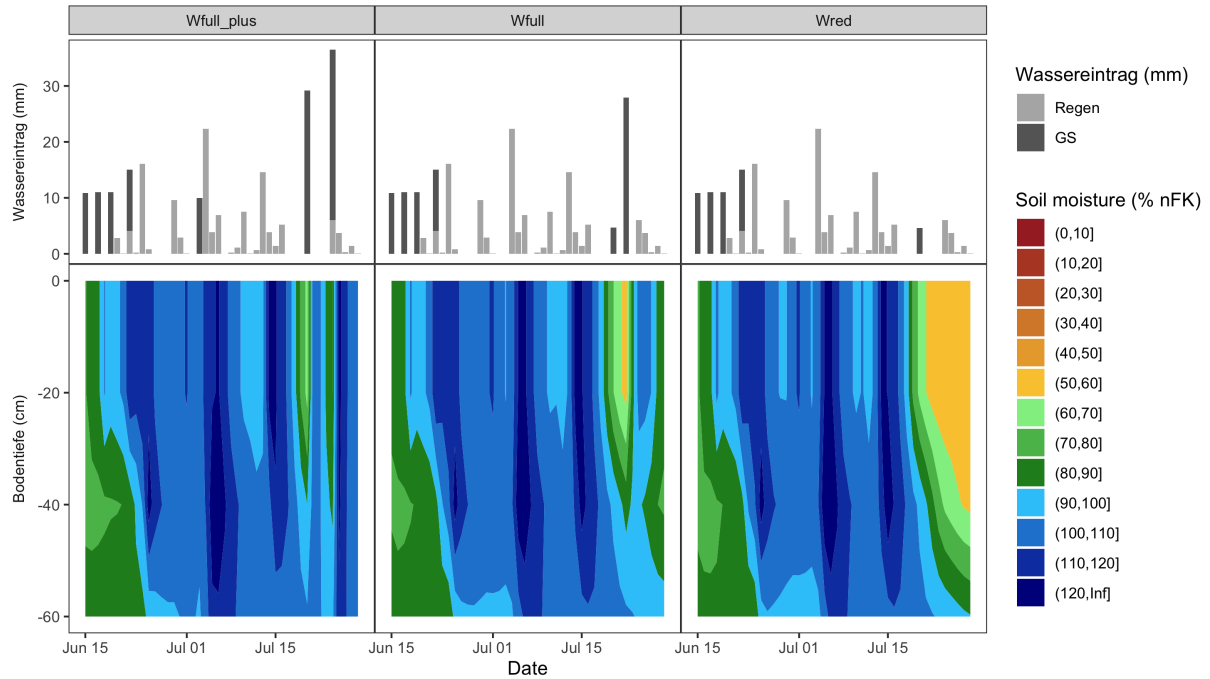
Bodenfeuchte und Wassereintrag für drei Behandlungen 2020, Satz 2, Feld 4a (*Paper VI über diese Daten)



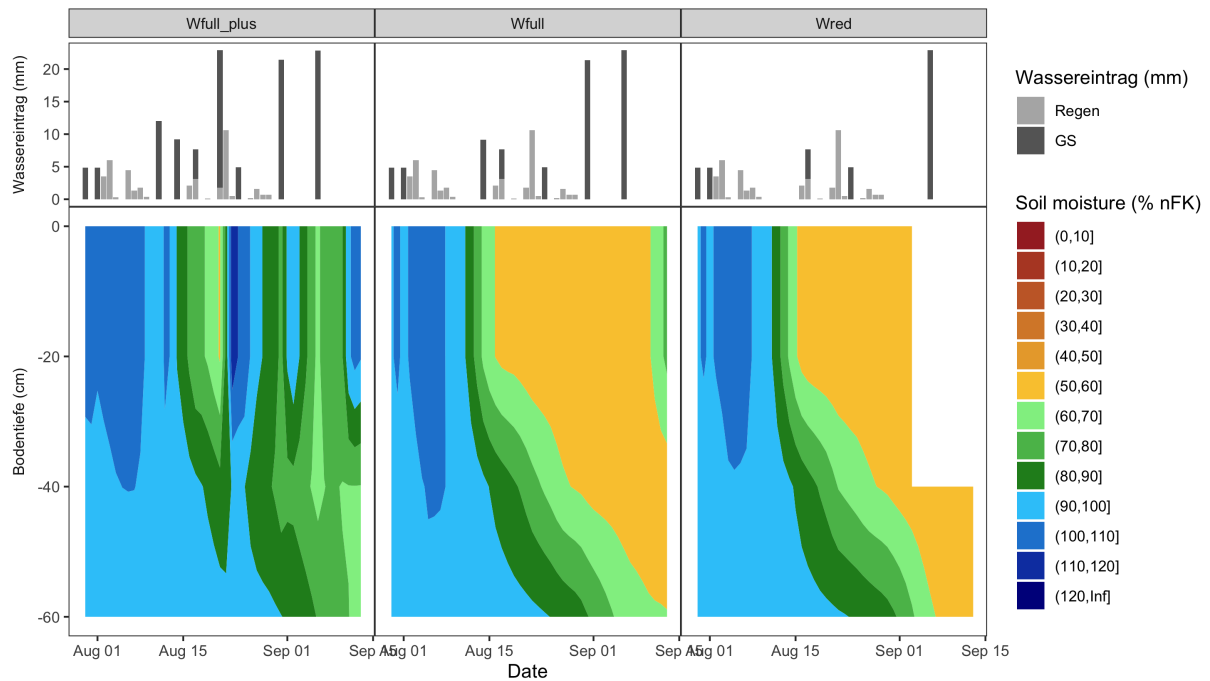
Bodenfeuchte und Wassereintrag für drei Behandlungen 2021, Satz 1, Feld 6



Bodenfeuchte und Wassereintrag für drei Behandlungen
2021, Satz 2, Feld 4b



Bodenfeuchte und Wassereintrag für drei Behandlungen
2021, Satz 3, Feld 6



#nFK-Tabelle speichern

```
#file2 <- "../data/derived_data/nfK_2021_Satz1.csv"
# file2 <- "../data/derived_data/nfK_2020_Satz2.csv"
# write.table(x = tensio, file = file2, sep = ";", dec = ".", row.names = FALSE)
```

#Bewässerung_mean-Tabelle speichern

```
#file1 <- "../data/derived_data/bewaessering_mean_2021_S1.csv"  
#write.table(x = bewaessering_mean, file = file1, sep = ";", dec = ".", row.names = FALSE)
```