

TRABALHO FINAL DA DISCIPLINA BIE5782 - FUNCAO TRANSMISSIVIDADE

Arquivo da Função

[funcao_transmissividade.r](#)

[ajuda_transmissividade.txt](#)

Arquivos com dados brutos dos ensaios

[slug_1503.txt](#)

[slug_ergomat.txt](#)

[pat.txt](#)

Para rodar a função com esses dados usar os parâmetros especificados na Página de Ajuda

Código da Função

```
# Funções de suporte utilizadas para cálculo da transmissividades
install.packages("lubridate", dependencies=TRUE)

library(lubridate)

transmissividade=function(dados,diam,rinf,L){

#funcao que calcula a transmissividade de uma porcao de um aquifero e
retorna os graficos de interpretacao e o resultado para o usuario

#os dados de entrada dessa funcao sao: dados (conjunto de dados numéricos
com a leitura do sensor de pressão (PSI), temperatura (°C) e data e hora);
diam (diâmetro do poço onde foi realizado o ensaio (mm)); rinf (raio de
influência do ensaio (m)); e L (comprimento do intervalo ensaiado (m)).

#####-----Etapa de verificação dos parâmetros e ajuste dos dados-
-----#####

if(class(diam) != "numeric" | diam<25.4 | diam>305){

  stop("valor não numérico ou diâmetro fora da faixa de 25.4 a 305 mm")}
#teste lógico para verificar se o parâmetro de entrada diâmetro (diam) está
dentro das especificações corretas

if(class(rinf) != "numeric" | class(rinf) == "integer" | rinf<1 |
rinf>1000){
```

```
stop("rinf precisa ser numero inteiro entre 1 e 1000")} #teste lógico para
verificar se o parâmetro de entrada raio de influência (rinf) está dentro
das especificações corretas

if(class(L) != "numeric" | L <= 0 ){

  stop("L precisa ser numérico e > 0")} #teste lógico para verificar se o
parâmetro de entrada comprimento (L) está dentro das especificações corretas

  x<-read.table(dados,header=TRUE,sep=";",dec=".",na.strings =
c("","NA","NaN")) #lê o arquivo com os dados brutos na pasta de trabalho
atribuindo NA para os campos vazios, com NA e NaN (esse último ocorre, em
geral, quando há problemas na comunicação com os sensores)

x<-subset(x=x,select=c(TIMESTAMP,Int_Temp,Int_Psi)) #seleciona apenas os
campos de interesse para a interpretação do ensaio slug

for(i in 1:3){ #contador para fazer o teste lógico de quantos NAs existem em
cada coluna de dados

  y<-length(subset(x=x[,i],subset=x[,i]=="NA")) #conta e guarda no objeto y
o numero de NAs por coluna

  if(y>0){ #teste lógico para ver se o número de NAs é maior que zero

    cat(paste("objeto contém NA em", y, "registros na coluna de
dados",colnames(x)[i],"\n")) #mensagem com o número de NAs por coluna

  }else{

    cat(paste("objeto não contém NA na coluna de
dados",colnames(x)[i],"\n"))} #mensagem dizendo que não há NAs nesta coluna
}

x<-na.omit(x) #retira todos os registros de dados onde existem NAs
mencionados nos registros anteriores

z=!is.na(parse_date_time(x$TIMESTAMP,orders="dmy_HMS")) #verifica se existem
datas no campo x$TIMESTAMP fora do padrão ou com erro armazenando tudo que
for falso no objeto y

if(length(z[FALSE])>0){ #caso exista 1 ou mais campos de data fora do padrão
(FALSE) a função retorna a mensagem "campo data está fora do formato", caso
contrário "o campo data está correto"

  stop("campo data TIMESTAMP está fora do formato")
}

x$TIMESTAMP<-dmy_hms(x$TIMESTAMP) #converte o campo de data hora
```

```
(x$TIMESTAMP) de fator para data (POSIXct) com a função dmy_hms do pacote
lubridate

if(!(class(x$Int_Psi) == "numeric")){ #Controle de fluxo baseado em material
de aula do curso BIE5782

  stop( "coluna de dados Int_Psi não é um vetor numérico")}

if(!(class(x$Int_Temp) == "numeric")){ #Controle de fluxo baseado em
material de aula do curso BIE5782

  stop( "coluna de dados Int_Temp não é um vetor numérico")}

cat(paste(length(x$Int_Psi)," número total de registros disponíveis para
análise da transmissividade","\n","\n"))

#####-----Etapa de cálculo da transmissividade-----
#####

(diff(r.temp<-range(x$Int_Temp))) #variação da temperatura ao longo do
ensaio

(m.temp<-mean(x$Int_Temp)) #média da temperatura ao longo do ensaio
utilizada para calcular a densidade da água

(d.agua<-0.0000503*m.temp^3-0.00858*m.temp^2+0.0749*m.temp+1000) #densidade
da água em função da temperatura (Kg/m3)

(pe.agua<-d.agua*9.81) #cálculo do peso específico da água (densidade *
aceleração da gravidade em m/s2)

(mH20<-6894.75729/pe.agua) #fator de conversão da pressão em PSI para mH20

x$Int_mH20<-x$Int_Psi*mH20 #cria um campo na tabela de dados x com os
valores de pressão do sensor convertidos para mH20

quartz() #Abre a janela de gráficos no quartz para uma melhor visualização

par(mfrow=c(2,3)) #Divide a janela de gráfico em 6 espaços para mostrar
todas as etapas de evolução to tratamento e análise dos gráficos

plot(x$Int_mH20~x$TIMESTAMP,main="Todos os dados gerados em
campo",xlab="Tempo (hh:mm)",ylab="Pressão (mH20)") #plota o gráfico com
todos os dados do ensaio já transformados de PSI para mH20 em função da
Temperatura

ensaio<-subset(x=x,select=c(TIMESTAMP,Int_mH20)) #subset dos conjuntos de
dados que são realmente utilizados para a interpretação do ensaio (somente o
tempo e a variação da pressão em mH20)

for(i in 1:(length(ensaio$Int_mH20)-1)){ #ciclo for para identificar o
```

momento zero (H0) e o momento do início do ensaio (H1)

```
if((ensaio[i,2]+0.15) > (ensaio[i+1,2])){ #o critério para o início do ensaio foi uma variação positiva de 0.15 metros na pressão, indicando uma mudança brusca que não é característica do meio mas sim de uma intervenção
```

```
}else{
```

```
  H1<-ensaio[i+8,1:2] # momento em que o ensaio começa, termina a oscilação de pressão inicial de abertura/fechamento de válvula e o fluxo passa a ser controlado pelo meio
```

```
  H0<-ensaio[i-30,1:2]} # momento em que a pressão de água está em equilíbrio ou mais próximo disso  
}
```

```
ensaio<-  
subset(x=ensaio,subset=ensaio$TIMESTAMP>=H1[1,1],select=c(TIMESTAMP,Int_mH20)) #subset do intervalo de dados que são utilizados na interpretação do ensaio, tempo a partir do H1 até o final dos dados
```

```
plot(ensaio$Int_mH20~ensaio$TIMESTAMP,main="Dados após início do ensaio de slug",xlab="Tempo (hh:mm)",ylab="Pressão (mH20)") #plot do intervalo de dados do ensaio
```

```
ensaio$dH<-ensaio$Int_mH20-H0$Int_mH20 #cria um novo campo no objeto ensaio que recebe a variação de pressão em mH20
```

```
ensaio$dHdH1<-ensaio$dH/(H1$Int_mH20-H0$Int_mH20) #cria um novo campo no objeto ensaio que recebe a normalização dos dados do ensaio, ou seja, a variação da pressão com relação à variação máxima observada (H1-H0)
```

```
ensaio$ln.dHdH1<-log(x=ensaio$dHdH1) #cria um novo campo no objeto ensaio que recebe o calculo de log normal do valor normalizado do ensaio (ln(dH/dH1))
```

```
is.na(ensaio$ln.dHdH1)<-sapply(ensaio$ln.dHdH1,is.infinite) #substitui os valores de infinito, obtidos na operação anterior, por NA para que não haja erro na hora de realizar o modelo linear dos dados de "ensaio$ln.dHdH1". Esta solução foi retirada do video publicado por Sarveshwar Inani em out/2015 disponível em https://www.youtube.com/watch?v=a\_vCQC0jdpk
```

```
ensaio$Ts<-1+(ensaio$TIMESTAMP-ensaio$TIMESTAMP[1]) #cria um novo campo no objeto "ensaio" que recebe o tempo de duração do ensaio em segundos
```

```
plot(ensaio$ln.dHdH1~ensaio$Ts,main="Dados do ensaio após normalização",xlab="Tempo (s)",ylab="ln.dH/dH1") #plota os dados normalizados do ensaio
```

```
lm.ensaio<-lm(ensaio$ln.dHdH1 ~ ensaio$Ts) #gera o modelo linear dos dados
normalizados (ensaio$ln.dHdH1) do ensaio em função do tempo em segundos
(ensaio$Ts)

abline(lm.ensaio,col="red") #plota no gráfico dos dados normalizados do
ensaio a reta do modelo linear

r<-setNames(data.frame(matrix(ncol = 5, nrow = 1)),
c("min","max","dif","cincP","setentaP")) #cria um dataframe no objeto r com
as variáveis que são utilizadas para fazer a interpretação do ensaio. Para
isso foi utilizado o código disponível em:
https://stackoverflow.com/questions/32712301/create-empty-data-frame-with-co
lumn-names-by-assigning-a-string-vector/32712555

r$min=min(ensaio$Int_mH20) #menor valor de pressão do ensaio

r$max=max(ensaio$Int_mH20) #maior valor de pressão do ensaio

r$dif=r$max-r$min #diferença de pressão observada

r$cincP=r$max-r$dif*0.05 #corte de 5% do total de variação da pressão nos
primeiros dados do ensaio que estão sujeitos a não idealidades do poço ou
tempo de equilíbrio

r$oitentaP=r$max-r$dif*0.8 #corte dos 20% do total de variação da pressão
no final do ensaio que estão sujeitos estabilização e tem um padrão de
função logarítmica

ensaio.t<-subset(x=ensaio,subset=ensaio$Int_mH20<r$cincP &
ensaio$Int_mH20>r$oitentaP) #faz um subset dos dados retirando os 5% da
variação inicial e os 20% finais

ensaio.t$r2<-NA #cria o campo r2 no objeto ensaio para receber o
coeficiente de coerrelação que será utilizado para selecionar o melhor
conjunto de dados na obtenção da inclinação da reta, que por sua vez será
utilizada para o cálculo da transmissividade

for(i in 3:length(ensaio.t$Ts)){ #entra em um ciclo for para calcular o
modelo linear do conjunto de dados do ensaio a partir do 3º registro, porque
o modelo linear de dois pontos da r2 de 1, e obter o r2 para cada intervalo
de dado em função do tempo. Os valores de r2 são armazenados no objeto
ensaio$r2

  lm.ensaio.t<-lm(ensaio.t$ln.dHdH1[1:i] ~ ensaio.t$Ts[1:i])

  ensaio.t$r2[i]<-summary(lm.ensaio.t)$r.squared #retira o r2 usando uma
indexação do summary. Essa linha foi retirada de
http://www.r-tutor.com/elementary-statistics/simple-linear-regression/coeffi
cient-determination
```

```
}

m.r2=max(ensaio.t$r2,na.rm=TRUE) #encontra o maior valor de r2 e guarda no
objeto m.r2

rows<-which(grepl(m.r2,ensaio.t$r2)) #usa a função which com grepl para
selecionar o registro que apresenta o maior r2 (armazenado em m.r2) e salva
esse registro no objeto rows

ensaio.t<-ensaio.t[1:rows,1:7] #faz uma seleção dos dados compreendidos no
intervalo que obteve o melhor r2 e armazena no objeto ensaio

plot(ensaio.t$ln.dHdH1 ~ ensaio.t$Ts,main="Dados para cálculo da
transmissividade",xlab="Tempo (s)",ylab="ln.dH/dH1") #faz um plot dos
resultados do log normal da carga normalizada em função do tempo

lm.ensaio.t<-lm(ensaio.t$ln.dHdH1 ~ ensaio.t$Ts) #cria um modelo linear
para os resultados do log normal da carga normalizada em função do tempo

abline(lm.ensaio.t,col="blue") #traça a reta no gráfico gerado do log normal
da carga normalizada em função do tempo

plot(ensaio$ln.dHdH1~ensaio$Ts,main="Interpretação do ensaio de
slug",xlab="Tempo (s)",ylab="ln.dH/dH1") #plota os dados normalizados do
ensaio

abline(lm.ensaio,col="red") #plota no gráfico dos dados normalizados do
ensaio a reta do modelo linear com todos os dados do ensaio

abline(lm.ensaio.t,col="blue") #traça a reta no gráfico gerado do log normal
da carga normalizada em função do tempo com os dados compreendidos entre o
melhor R2

plot(0, type='n', axes=FALSE, ann=FALSE) #plota um gráfico em branco onde
será colocada a legenda das figuras

legend(x=0.58, y=1, legend=c("lm melhor r2","lm todo ensaio"),
title=expression(bold("Legenda")), col=c(4,2), lty=1, bty=1, cex=1.2)
#insere a legenda das figuras

inclinacao.reta<-lm.ensaio.t$coefficients[2] #guarda a inclinação da reta
do modelo linear acima no objeto inclinacao.reta

shape.factor=(2*pi*L)/(log(rinf/(diam/2000))) #calcula o shape factor do
poço com os dados que foram fornecidos na função Transmissividade

valor.T<-inclinacao.reta*(pi*(diam/1000)^2/4)*shape.factor*L #calcula o
valor da Transmissividade e armazena no objeto valor.T
```

```
resultados<-setNames(data.frame(matrix(ncol = 6, nrow = 1)), c("T
(m2/s)","R2","rinf (m)","diam (mm)","t.ensaio","h.max (m)") #cria um
dataframe no objeto resultados com os resultados que são apresentados como
saída da função para o usuário. Para isso foi utilizado o código disponível
em:
https://stackoverflow.com/questions/32712301/create-empty-data-frame-with-co
lumn-names-by-assigning-a-string-vector/32712555

resultados[,1]=valor.T #Valor da transmissividade

resultados[,2]=round(m.r2,3) #melhor r2 com até 3 casas decimais, utilizado
para definir o conjunto de dados a ser utilizado no modelo linear

resultados[,3]=rinf #raio de influência fornecido pelo usuário da função

resultados[,4]=diam #diâmetro do poço fornecido pelo usuário da função

resultados[,5]=max(ensaio$Ts) #tempo de ensaio utilizado para o cálculo da
transmissividade

resultados[,6]=r$dif #deslocamento máximo da coluna de água durante o
ensaio

return(resultados) #reotrna a tabela dos resultados para o usuário
}
```

Página de Ajuda

Transmissividade
Documentation

package:unknown

R

CALCULA A TRANSMISSIVIDADE A PARTIR DE ENSAIOS HIDRAULICOS TIPO SLUG

Description:

Função para calcular a transmissividade de aquíferos ou trechos discretos destes a partir de ensaios hidráulicos do tipo slug (Butler, 1998). A solução analítica utilizada é Hvorslev (1951). A função abre o arquivo de dados brutos gerados em campo pelos sensores e retorna o valor da transmissividaae.

Usage:

transmissividade (dados, diam, rinf, L)

Arguments:

dados: tabela no formato txt contendo os dados do ensaio tipo slug para o calculo

da transmissividade. Os registros da tablea devem ser dados numéricos contendo os

registros do sensor de pressão (em PSI), temperatura ($^{\circ}\text{C}$) e data/hora.

diam: número com intervalo de $25.4 < \text{diam} < 305$ correspondente ao diâmetro do poço

onde foi realizado o ensaio (unidade considerada em mm).

rinf: número inteiro no intervalo $1 < \text{rinf} < 1000$ representando o raio de influência

do ensaio (unidade considerada em m).

L: número maior que zero correspondente ao comprimento do intervalo ensaiado

(unidade considerada em m).

Details:

Os “dados” de entrada podem ter diferentes campos e nomes desde que sejam respeitados

os nomes das três colunas que são utilizadas para o calculo da transmissividade

(TIMESTAMP, Int_Temp, Int_Psi). O método de interpretação de Hvorslev contempla os

casos de aquíferos confinados não drenantes e aquíferos livres ensaiados com poços

afogados.

Value:

Gráfico com todos os dados importados da tabela “dados” com a pressão já convertida de

PSI para mH2O

Gráfico com os dados após o início do ensaio

Gráfico com os dados normalizados após o início do ensaio e a reta da regressão linear

de todo esse conjunto de dados

Gráfico com o melhor coeficiente de correlação no intervalo de 5 a 80% dos dados de

variação da pressão no ensaio e a reta da regressão linear cuja inclinação é utilizada

para o cálculo da transmissividade

Gráfico com os dados normalizados após o início do ensaio e as retas da regressão linear de todo conjunto de dados e com o melhor coeficiente de correlação

Tabela com o resultado do cálculo da transmissividade e os principais parâmetros utilizados

e indicadores de qualidade da interpretação para o usuário:

- Transmissividade em m²/s
- coeficiente de correlação do conjunto de dados utilizado (R²)
- raio de influencia fornecido pelo usuário (rinf)
- diâmetro do poço
- tempo do ensaio utilizado na interpretação da transmissividade (t.ensaio)
- deslocamento máximo observado no ensaio (h.max)

Warning:

Caso algum dos argumentos inserido esteja fora das especificações a função não é executada

e os seguintes avisos são gerados pela função:

- valor não numérico ou diâmetro fora da faixa de 25.4 a 305 mm
- rinf precisa ser numero inteiro entre 1 e 1000
- L precisa ser numérico e > 0
- campo data TIMESTAMP está fora do formato
- coluna de dados Int_Psi não é um vetor numérico
- coluna de dados Int_Temp não é um vetor numérico

Na etapa de verificação e ajuste dos dados, uma mensagem automática é gerada dizendo o número

de registros excluídos por apresentarem valor NA

Ao final da função uma mensagem automática é gerada dizendo o número de registros utilizados

na interpretação da transmissividade

Author(s):

Marcos Bolognini Barbosa
e-mail: marbbar@usp.br

References:

Butler, J.J., Jr., Duffield, G.M. and D.L. Kelleher, 2011. Field Guide for Slug Testing and Data Analysis, Midwest Geosciences Group Press.

Hvorslev, M.J., 1951. Time Lag and Soil Permeability in Ground-Water Observations, Bull.

No. 36, Waterways Exper. Sta. Corps of Engrs, U.S. Army, Vicksburg, Mississippi, pp. 1-50.

Examples:

```
transmissividade(dados="Slug_1503.txt",diam=152.24,rinf=30,L=2)
```

```
transmissividade(dados="Slug_Ergomat.txt",diam=50.8,rinf=30,L=2)
```

```
transmissividade(dados="Pat.txt",diam=50.8,rinf=30,L=2)
```

From:

<http://ecor.ib.usp.br/> - **ecoR**

Permanent link:

http://ecor.ib.usp.br/doku.php?id=05_curso_antigo:r2018:alunos:trabalho_final:marbbar:transmissividade 

Last update: **2020/08/12 06:04**