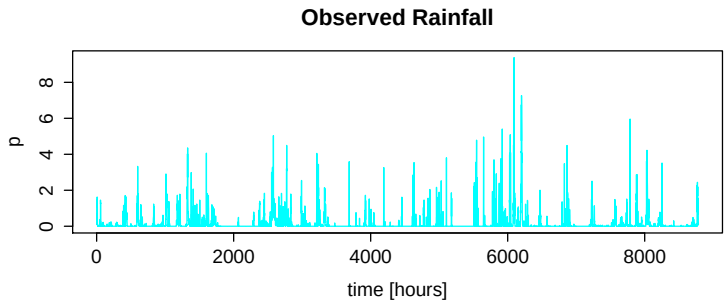
March 20, 2015

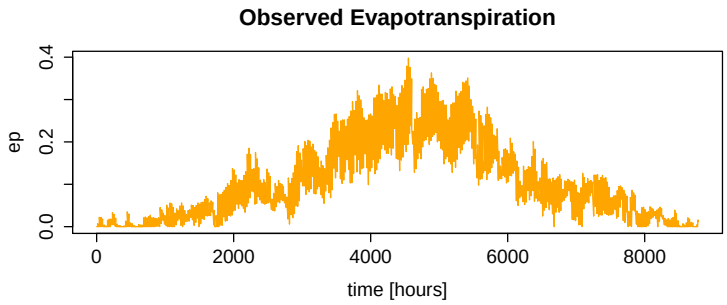
The results below are generated from an R script.

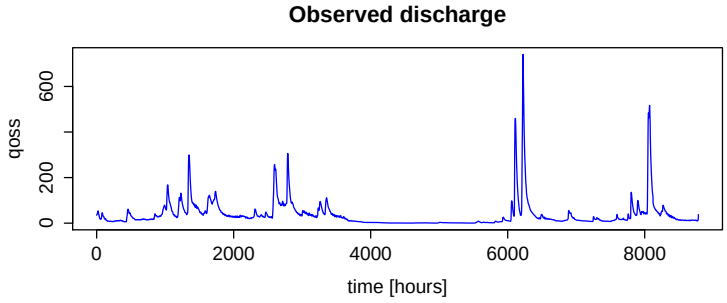
```
      rm(list=ls())  ## Remove Objects

      ################################################
      ## Set the working directory
      ################################################

      setwd("/home/emanuele/Desktop/Hymod")

      ################################################
      ## Files in the folder
      ################################################

      dir()
```

```
## [1] "datisecchia.dat" "Hymod.R"
```

```
      ################################################
      ## Load the files (OBSERVED Q, P and EP)
      ################################################

      dati<-read.table("datisecchia.dat",header = TRUE,sep="")
      p<-dati[,2]
      ep<-dati[,3]
      qoss<-dati[,1]
      rm(dati)

      par(mfrow=c(3,1))
      plot(p,col="cyan",ty="l",main="Observed Rainfall",xlab="time [hours]")
      plot(ep,col="orange",ty="l",main="Observed Evapotranspiration",xlab="time [hours]")
      plot(qoss,col="blue",ty="l",main="Observed discharge",xlab="time [hours]")
```








```
# dev.off()  ## close the graphical device

################################################
## Hymod
################################################
hymod=function(par)
{
  areab=1314000000
  tstep=3600         # 3600 for hourly data; 86400 for daily data
  ndeltat=8784       # be careful to adjust it as needed
  qi=1
  fatconv=(1/1000/tstep)*areab

  # parameters #

  cmax=par[1]
  beta=par[2]
  alfa=par[3]
  kslow=par[4]
  kquick=par[5]
```

```
# initialisation of variables #

w2=0
c1=0
wslow=qi/(kslow*fatconv)
wquick=rep(0,3)
er1=rep(0,ndeltat)
er2=rep(0,ndeltat)
er=rep(0,ndeltat)
e=rep(0,ndeltat)
qt=rep(0,ndeltat)

# computation loop #

for (i in 1:ndeltat)

{

  w1=w2
  c1=cmax*(1-((1-((beta+1)*w1/cmax))^(1/(beta+1))))
  er1[i]=max((p[i]-cmax+c1),0)
  c2=min(c1+p[i],cmax)
  w2=(cmax/(beta+1))*(1-((1-c2/cmax)^(beta+1)))
  er2[i]=max((c2-c1)-(w2-w1),0)
  e[i]=(1-(((cmax-c2)/(beta+1))/(cmax/(beta+1))))*ep[i]
  w2=max(w2-e[i],0)
  er[i]=er1[i]+er2[i]

  # Subdivision of the surface runoff #

  uquick=alfa*er[i]
  uslow=(1-alfa)*er[i]

  # Slow flow #

  wslow=(1-kslow)*wslow+(1-kslow)*uslow
  qslow=(kslow/(1-kslow))*wslow


  # Quick flow #

  qquick=0
  for (j in 1:3)
  {
    wquick[j]=(1-kquick)*wquick[j]+(1-kquick)*uquick
    qquick=(kquick/(1-kquick))*wquick[j]
    uquick=qquick
  }

  # Total flow #

  qt[i]=(qslow+qquick)*fatconv

}
```

```
  # Drawing the plot
  #plot(qt,type="l",ylim=c(0,max(c(max(qt),max(qoss)))),col="red",lwd=2)
  #lines(qoss)

  # Return the simulated discharge
  #return(qt)

  ## Compute the MSE
  mse<-mean( (qt-qoss)^2)
  # Return the MSE value
  return(mse)

}



################################################
## Optimization procedure
################################################
# install.packages("DEoptim")
#library("DEoptim")
#
#lower_val=c(10,0.1,0.4,0.001,0.01)
#upper_val=c(700,2,0.99,0.1,0.99)
#
#optim_param_MSE<-DEoptim(fn=hymod,
#                  lower=lower_val, upper=upper_val,
#                  DEoptim.control(itermax = 100, trace = TRUE))
#
# the best set of parameters found
# optim_param_MSE$optim$bestmem
#  the value of fn corresponding to the best set of parameters
# optim_param_MSE$optim$bestval
```