

```

"nlrq"<-
function(model, t, k = 2, theta = .5, big=1e+20, nit.max = 100,
        eps = 1e-07, beta = 0.97)
{
#function to compute nonlinear rq estimate
#   t is the initial value of the unknown parameter
#   model is a user-provided function which returns components
#       f=(f_i (x_i , t)
#       J=(grad f_i )
#   theta is the desired quantile
#   k is the number of Meketon steps per iteration
#   eps and eta are tolerance parameters
#
#function returns
#   coef is the value of the parameter at the solution
#   obj is the value of the objective function at the solution
#   nit is the number of "Meketon steps" taken
  m <- model(t)
  n <- length(m$f)
  w <- rep(0, n)
  snew <- sum(rho.rq(m$f,theta))
  sold <- big
  nit <- 0
  while(sold - snew > eps & nit < nit.max) {
    z <- mek.rq(m$J, m$f, k, w, theta=theta, int = F,
               eps = eps, beta = beta)

    step <- z$coef
    t0 <- t
    l <- nlminb(start = 1, objective = model.step.rq, lower = 0,
               upper = 1, model = model, t0 = t, theta=theta,
               step = step)$parameters
    t <- t0 + l * step
    m <- model(t)
    sold <- snew
    snew <- sum(rho.rq(m$f,theta))
    w <- lsfit(m$J, z$w, int = F)$resid
    w1 <- max(pmax(w,0))
    if(w1>theta)
      w <- w*theta/(w1 + eps)
    w0 <- max(pmax(-w,0))
    if(w0>1-theta)
      w <- w*(1-theta)/(w0 + eps)
    print(c(t, l, sum(rho.rq(m$f,theta))))
    nit <- nit+1
  }
  return(coef=t,obj=snew,nit=nit)
}

"rho.rq"<-
function(u,theta){theta*pmax(u,0)+(theta - 1)*pmin(u,0)}
"mek.rq"<-
function(x, y, kmax = 1000, w, theta=.5, int = T, big=1e+20,
        eps = 1e-06, beta = 0.97)
{
  if(int == T)
    x <- cbind(1, x)
  yw <- big
  k <- 1
  while(k <= kmax & yw - crossprod(y, w) > eps) {
    d <- pmin(theta - w, 1 - theta + w)
    z <- lsfit(x, y, d^2, int = F)
    yw <- sum(rho.rq(z$resid,theta))
    k <- k + 1
    s <- z$resid * d^2
  }
}

```

```
        alpha <- max(eps, pmax(s/(theta - w), -s/(1 - theta + w)))
        w <- w + (beta/alpha) * s
    }
    coef <- z$coef
    return(coef, w)
}
"model.step.rq"<-
function(lambda, t0, step, model, theta)
{
    sum(rho.rq(model(t0 + lambda * step)$f, theta))
}
```