

```

# function to compute the Gaussian for matrix/vector
G <- function(f, g, x, y, h) {
  exp(-(x - y - f(y)*h)^2/(2*g(y)^2*h))/(abs(g(y))*sqrt(2*pi*h))
}

dtq <- function(f, g, h, k, T, L, init, final) {
  numsteps = ceiling(T/h)      # number of DTQ steps
  zvec = seq(-L, L, by = k)    # grid z

  # creating the matrix A
  zmat = replicate(length(zvec), zvec)
  A = G(f, g, zmat, t(zmat), h)

  # pdf after first timestep
  approxpdf = k * (as.matrix(G(f, g, zvec, init, h)))

  # (n-2) iterative steps
  for (i in c(2:numsteps-1))
    approxpdf = k*(A %*% approxpdf)

  # pdf at final timestep
  approxpdf = k * (as.matrix(G(f, g, final, zvec, h)))
}

```