Flux Calculation with the Quantized Eddy Accumulation Method

Anas Emad *

2023-12-27

This document outlines a simple calculation of the fluxes using the quantized eddy accumulation (QEA) method with error diffusion. We generate some synthetic data and simulate the applicaiton of the QEA method then calculate the fluxes

Setup

```
# We define the quantization function as described in the accompanying paper
quantize_w <- function(w,
                       quantize_threshold,
                       w_full_scale_value) {
    residual_error_updraft   <- 0
    residual_error_downdraft <- 0
    N <- length(w)
    w_qua <- rep(0, N)

    for (i in seq_along(w)) {
        # Error diffusion is carried out within the respective wind direction
        if (w[i] > 0) {
            # Add error from previous step to modified w
            w_mod <- w[i] - residual_error_updraft
            # Quantize modified wind
            w_qua[i] <- ifelse(abs(w_mod) > quantize_threshold,
                               sign(w_mod)*w_full_scale_value, 0)
            # Calculate residual error
            residual_error_updraft <- w_qua[i] - w_mod
        } else {
            w_mod <- w[i] - residual_error_downdraft
            w_qua[i] <- ifelse(abs(w_mod) > quantize_threshold,
                               sign(w_mod)*w_full_scale_value, 0)
            residual_error_downdraft <- w_qua[i] - w_mod
        }

    }
    return(w_qua)
}

# We define the biased covariance function

bcov <- function (x,y) {
    N <- length(x)
```

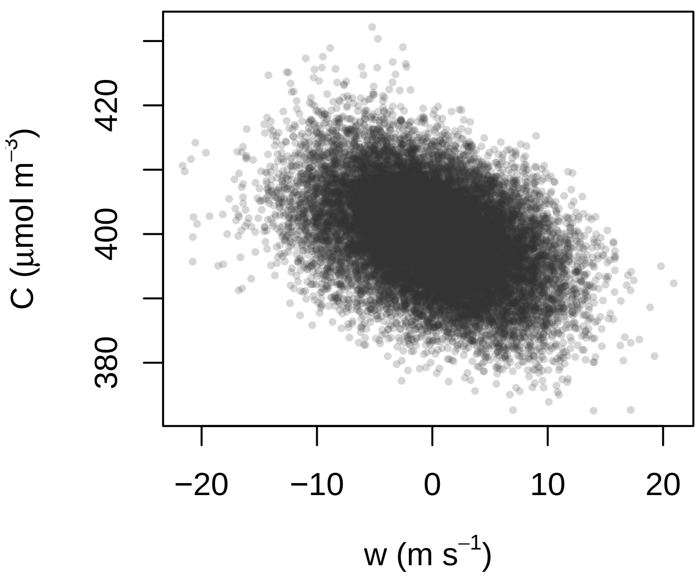
*Email: anas.emad@uni-goettingen.de

```
    cov(x,y, use = "na.or.complete")*(N-1)/N
}
```

Data Generation

We first generate some synthetic data for variables `w` (wind) and `C` (scalar concentration). The generated time series is equivalent to 30 minutes of 10 Hz sampled data.

```
N  <- 18000

set.seed(17)
ar_params  <- list(ar = c(0.993499, 0.017448, -0.218639, 0.136650))

w <- arima.sim(n = N, ar_params, sd = 2)

set.seed(18)
C <- arima.sim(n = N, ar_params, sd = 2.5) - w * 0.6 + 400

# We consider a nonzero mean of vertical wind in the range of 0.02 m
w <- w - mean(w) + 0.02

plot(w, C, pch = 20, col = rgb(0.2, 0.2, 0.2, 0.2), cex = 0.6,
     xlab = expression("w (m s"^-1*")"),
     ylab = expression("C (" * mu * "mol m"^-3*")"))
```



Quantization of wind

Next, we quantize the wind data and calculate the quantization error.

```
quantization_threshold <- sd(w) * 2.5
w_full_scale_value     <- sd(w) * 4

wq                     <- quantize_w(w, quantization_threshold, w_full_scale_value)
wqr                    <- w - wq
```

Flux Calculation

Quantities Needed for Flux Calculation

Below are the quantities required for calculating the flux

```
Cq_up_mean   <- mean(C[wq>0])   # Mean concentration in updraft reservoir (umol m^-3)
Cq_down_mean <- mean(C[wq<0])   # Mean concentration in downdraft reservoir
wq_up_mean   <- mean(wq[wq>0])  # Mean wq when wq > 0 which equals w_full_scale_value (m s^-1)
wq_down_mean <- mean(wq[wq<0])  # Mean wq when wq < 0 which equals -w_full_scale_value
N            <- length(w)       # Total count of w samples
Nq           <- sum(wq!=0)      # Count of collected samples
Nq_up        <- sum(wq>0)       # Count of collected samples in updraft
Nq_down      <- sum(wq<0)       # Count of collected samples in downdraft
alpha_co2_q  <- bcov(C, abs(wq))/bcov(C, wq) # Alpha: atmospheric asymmetry coefficient
```

The estimation of alpha and its value is discussed in: (Emad and Siebicke, 2023).

Next, we calculate the flux and the quantization flux error, following the equations provided in the paper.

```
# First order approximation of the flux (Eq. 22 in the paper)
# Unit of the flux ( umol m^-2 s^-1 )
Flux1 <- Cq_up_mean*Nq_up/N*wq_up_mean +
         Cq_down_mean*Nq_down/N*wq_down_mean -
         (Cq_up_mean*Nq_up/Nq + Cq_down_mean*Nq_down/Nq)*mean(wq)

# Quantization flux error (covariance of the scalar with the quantization error)
Quantization_flux_error <- bcov(C, wqr)

# Error ratio of the flux
Quantization_flux_error/Flux1
```

```
## [1] -0.009970666
```

```
# Nonzero w correction (Eq 23 in the paper)
Nonzero_w_correction <- 1 / (1 - alpha_co2_q * mean(wq)/mean(abs(wq)))
```

We find the relative error due to quantization is lower than 1% of the flux.

Verification

We can verify the covariance is equal to the flux calculated using the QEA method and check the quantized flux only.

```
# Verify covariance equals flux calculated with QEA
all.equal(bcov(C, w), Flux1 * Nonzero_w_correction + Quantization_flux_error)
```

```
## [1] TRUE
```

```
# Verify quantized flux only
all.equal(bcov(C, wq), Flux1 * Nonzero_w_correction)
```

```
## [1] TRUE
```