# Minimum Detectable ICOS Flux

To compute the minimum detectable flux I'm following the procedure used by Parkin et.al 2012 https://acsess.onlinelibrary.wiley.com/doi/epdf/10.2134/jeq2011.0394. Parkin et.al. computed minimum detection limit parameters for 3 and 4 measurements per incubation. He used a monte-carlo simulation of drawing random ppm values and computing fluxes for each set of values. Theoretically all fluxes drawn from that distribution should be zero. He defined the minium detection limit as the 95% percentile of the distribution of zero fluxes, leaving a 5% chance of classifying a zero flux as a "real" flux.

Parkin used a 4 point flux calculation, however with the ICOS we get a few hundred measurements for each incubation. I repeated the Monte Carlo simulation using the larger number of measurements possible with the ICOS.

First I need to compute the standard deviation and CV of the system. Parkin used CV, I used standard deviation, since the standard deviation is the input parameter to the `rnorm` function. The `rnorm` function is used to generate a distribution with a specified mean and standard devation.

We measured "zero" fluxes in the field by installing the chambers over a metal plates so there should not be any gas released into the chamber.

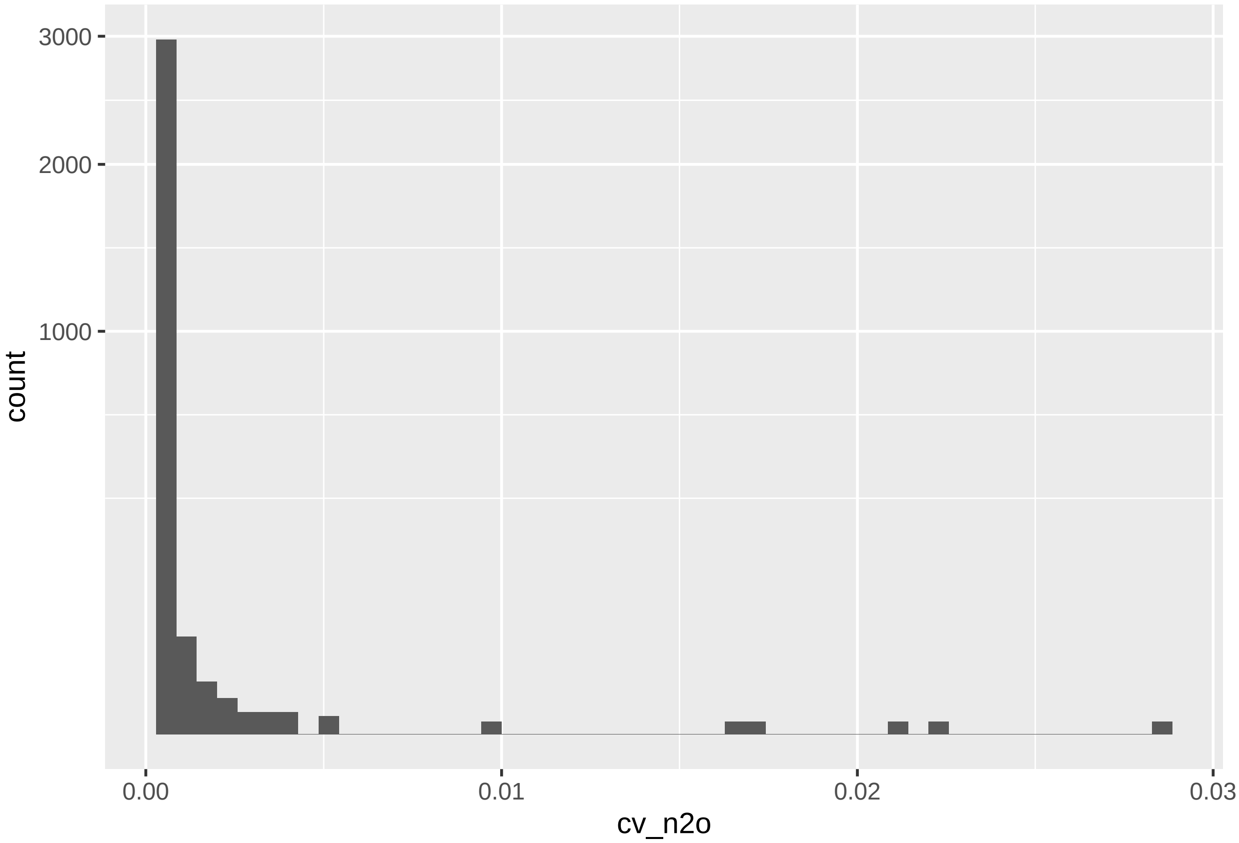
Taking all the zero fluxes (where the metal plate was installed), we compute the mean, standard deviation, median number of samples per incubation, and CV from each incubation.

```
points %>%
  group_by(id) %>%
  summarize(avg_n2o = mean(n2o_ppm), sd_n2o = sd(n2o_ppm), number = n()) %>%
  mutate(cv_n2o = sd_n2o/avg_n2o) %>%
  summarize(mean_n2o= mean(avg_n2o), mean_sd = mean(sd_n2o),
            max_cv = max(cv_n2o), min_cv = min(cv_n2o), mean_cv=mean(cv_n2o),
            median_n = median(number))
```

```
## # A tibble: 1 x 6
##   mean_n2o  mean_sd max_cv   min_cv  mean_cv median_n
##      <dbl>    <dbl>  <dbl>    <dbl>    <dbl>    <int>
## 1    0.350 0.000228 0.0285 0.000464 0.000652      210
```

Plotting the distribution of CV's a expected they look to be clustered on the low end.

```
points %>%
  group_by(id) %>%
  summarize(avg_n2o = mean(n2o_ppm), sd_n2o = sd(n2o_ppm)) %>%
  mutate(cv_n2o = sd_n2o/avg_n2o) %>%
  ungroup() %>%
  ggplot(aes(cv_n2o)) + geom_histogram(bins=50)  + scale_y_sqrt()
```



I define a function to draw 210 samples (which was the average number of sample points we have per incubation) from a random distribution (`rnorm(number, mean, std)`) with a mean of 0.350 ppm and a standard deviation of 0.000228 (computed from the zero flux data). Then I join it with a sequence (1..210) which would represent the second and divide by 30 to simulate minutes (we get one measurement every 2 seconds). I join the minute and ppm values into a data.frame and fit a linear model `ppm = a + b second` and extract the slope `b` component.

```
sim = function() {
  minute = seq(1,210)/30
  ppm=rnorm(210, 0.350, 0.000228)
  data = data.frame(minute, ppm)
  summary(lm(ppm ~ minute, data=data))$coefficients[[2,1]]
}
```

Running through one iteration of the function with the intermediate results, shows that the last result returns just the slope intercept.

```
minute = seq(1,210)/30
ppm=rnorm(210, 0.350, 0.000228)
data = data.frame(minute, ppm)
head(data)
```

```
##       minute       ppm
## 1 0.03333333 0.3498398
## 2 0.06666667 0.3504224
## 3 0.10000000 0.3503021
## 4 0.13333333 0.3501302
## 5 0.16666667 0.3502378
## 6 0.20000000 0.3502743
```

```
summary(lm(ppm ~ minute, data=data))$coefficients
```

```
##                  Estimate   Std. Error       t value  Pr(>|t|)
## (Intercept)  3.500158e-01 3.180605e-05  1.100469e+04 0.0000000
## minute      -3.619891e-07 7.841956e-06 -4.616056e-02 0.9632266
```

```
summary(lm(ppm ~ minute, data=data))$coefficients[[2 ,1]]
```

```
## [1] -3.619891e-07
```

Next I define a function to run a number of simulations using the `sim` function defined above, sorting the resulting slopes and taking the last number of the top 5% or 1% of the numbers to be the minimum detectable flux.

```
mdl = function(runs, significance, sim_function) {
  replicate(runs, sim_function()) %>%
  enframe(name='id', value='flux') %>%
  arrange(flux)  %>%
  # mutate(row_id =row_number()) %>%
  slice_head(n=runs * significance) %>%
  slice_tail()
}
```

Running the function 100000 times with a 5% and a 1% significance level. The minimum detectable slope at a 5% confidence level is (we have two tails so we use 0.025 as the cutoff)

```
mdl(100000, 0.025, sim)
```

```
## # A tibble: 1 x 2
##      id       flux
##   <int>      <dbl>
## 1 25504 -0.0000153
```

and at a 1% confidence level

```
mdl(100000, 0.005, sim)
```

```
## # A tibble: 1 x 2
##      id       flux
##   <int>      <dbl>
## 1 87862 -0.0000200
```

Since the minimum detectable slopes are symmetrical, the minimum detectable slope is around +/- 0.015 ppb/min at a 5% confidence level and +/- 0.02 ppb/min at a 1% confidence level.

Defining a function to compute flux in grams/ha/day

```
flux <- function(slope, molecular_weight, air_temperature, height) {
  ug_l_minute = (slope * molecular_weight * 1)/(0.0821 * (air_temperature + 273.15))
  area_m2 =  pi * 14.1^2/10000
  volume_l = pi * 14.1^2 * (height- 0.2) / 1000
  flux_ug_m_hr = (ug_l_minute * volume_l * 60)/area_m2
  flux_ug_m_hr * 0.01 * 24 # grams/ha/day
}
```

Assuming 20C and 30 cm chamber height we would get

```
flux(0.000015, 28, 20, 30)
```

```
## [1] 0.07488503
```

```
flux(0.000020, 28, 20, 30)
```

```
## [1] 0.0998467
```

For our chambers that would work out to an MDL of +/-74 mg/ha/day at a 5% confidence level and +/-99 mg/ha/day at a 1% confidence level.

The mean slopes for the zero flux samples (-0.0072 ppb/min) is less than the computed minimum detectable limit.

```
points %>%
  distinct(id, n2o_slope) %>%
  summarize(n2o_mean_slope = mean(n2o_slope))
```

```
## # A tibble: 1 x 1
##   n2o_mean_slope
##            <dbl>
## 1    -0.00000724
```

Which works out to about 18 mg/ha/day.

During the fall run-in period we observed 89 negative fluxes out of 1744 total fluxes. Of the negative fluxes 25 were below the MDL. However, if we filter out fluxes where the co2_r2 < 0.8 we end up with 4 negative fluxes below the MDL and if we filter out fluxes where the co2_r2 < 0.8 and the n2o_r2 < 0.5 we end up with no negative fluxes below the MDL.

Christiansen et. al. 2015 (https://doi.org/10.1016/j.agrformet.2015.06.004) proposed a method to compute the minimum detectable flux based on the published noise figure of the instrument used. He proposed $\frac{Aa}{tc}\frac{VP}{SRT}$ where $Aa$ is the analytic accuracy of the instrument, $tc$ is the closure time in hours $V$ is the chamber volume, $P$ the atmospheric pressure in Pa, $S$ the surface area of the soil, $R$ the gas constant and T the temperature in Kelvin.

Nickerson 2019 (https://eosense.com/wp-content/uploads/2019/11/Eosense-white-paper-Minimum-Detectable-Flux.pdf) proposed replacing instrument precsion with the standard error of the instrument precision by replacing the first element of the equation by $\frac{Aa}{tc\sqrt{(tc/p)}}$ where $p$ is the sampling period of the instrument.

Setting up the conditions for our chambers and computing with the Christiansen approach I get the following in mg/ha/day

```
# chamber closure in hours
tc = 210/30/60
# period in hours
p = 2/3600
# pressure in Pa
pressure = 101325
# temperature in K
temperature = 20 + 271
# surface area in m2
surface = 0.5 * 0.5
# gas constant for m3 Pa K-1 mol-1
R = 8.31446261815324
# volume in m2
volume = surface * 0.15
# precision in ppm
Aa = 2/1000

(Aa /tc ) * ((volume * pressure)/(surface * R * temperature)) * 28 * 24 / 10000 * 1000
```

```
## [1] 7.236577
```

while using the modified Nickerson approach I get:

```
(Aa / (tc * sqrt(tc/p))) * ((volume * pressure)/(surface * R * temperature)) * 28 * 24 / 10000 * 1000
```

```
## [1] 0.4993712
```

both are siginficantly lower than the Parkin approach and the actual zero flux measurements.

## Lab GC

we took 30 air samples to establish an error estimate

```
lab_data <- read_parquet("./cvresults.parquet") %>%
  filter(!str_detect(Sample, 'STD'))
```

lets get the standard deviations and CV's for N2O

```
lab_data %>%
  summarize(avg_n2o = mean(n2o_ppm), sd_n2o = sd(n2o_ppm), number = n()) %>%
  mutate(cv_n2o = sd_n2o/avg_n2o)
```

```
## # A tibble: 1 x 4
##   avg_n2o  sd_n2o number  cv_n2o
##     <dbl>   <dbl>  <int>   <dbl>
## 1   0.333 0.00481     30  0.0144
```

defining a lab simulation procedures drawing 4 samples for each incubation.

```
sim_lab = function() {
  minute = seq(1,4)*15
  ppm=rnorm(4, 0.3329667, 0.004810071)
  data = data.frame(minute, ppm)
  summary(lm(ppm ~ minute, data=data))$coefficients[[2,1]]
}
```

Again the minimum detectable slope at a 5% confidence level is (we have two tails so we use 0.025 as the cutoff)

```
mdl(100000, 0.025, sim_lab)
```

```
## # A tibble: 1 x 2
##      id      flux
##   <int>     <dbl>
## 1  4298 -0.000280
```

and at a 1% confidence level

```
mdl(100000, 0.005, sim_lab)
```

```
## # A tibble: 1 x 2
##      id      flux
##   <int>     <dbl>
## 1 97257 -0.000366
```

The minimum detectable slope is around +/- 0.28 ppb/min at a 5% confidence level and +/- 0.37 ppb/min at a 1% confidence level. Taking only 4 points results in a higher error estimate and a much larger MDL.

Assuming 20C and 30 cm chamber height and re-using our flux function we would get

```r
flux(0.000282, 28, 20, 30)
```

```
## [1] 1.407839
```

```r
flux(0.000369, 28, 20, 30)
```

```
## [1] 1.842172
```

For our chambers that would work out to an MDL of +/-1.4 g/ha/day at a 5% confidence level and +/-1.8 g/ha/day at a 1% confidence level.