Supplement of The Cryosphere, 18, 4379–4397, 2024
https://doi.org/10.5194/tc-18-4379-2024-supplement

The Cryosphere Open Access

Supplement of

Combining traditional and novel techniques to increase our understanding of the lock-in depth of atmospheric gases in polar ice cores – results from the EastGRIP region

Julien Westhoff et al.

Correspondence to: Julien Westhoff (julien.westhoff@nbi.ku.dk)

1 $\delta^{15}N$ isotopic ratios

Table 1: EGRIP firn air 2018, LSCE inert gas data (selected data only). Measured Oct 2019 on the Delta V by A. Orsi, without CO2 removal.

Depth	d15N
5	0.094
11.78	0.113
17.91	0.073
33.01	0.170
43.44	0.245
48.2	0.248
54.07	0.277
57.65	0.287
59.18	0.288
60.88	0.298
62.92	0.294
64.22	0.288
65.84	0.281
66.61	0.266

2 CO_2 mixing ratios

Table 2: EGRIP firn air 2018, CO_2 data. Correspondence: T. Blunier, University of Copenhagen, Denmark.

Depth	CO2
5.12	403
7.83	403
11.78	402
17.91	400
23.25	399
28.05	397
33.01	396
38.04	395
43.44	394
48.2	393
51.8	392
54.07	392
55.75	392
57.65	389
59.18	384
60.86	373
62.92	359
64.22	348
65.84	337

3 CH_4 mixing ratios

Table 3: EGRIP 2018 CH4. Measured at IGE by OF-CEAS 08-10/01/2019. Referenced to NOAA standards. StDev is the uncertainty on experimental data. Correspondance: xavier.fain@univ-grenoble-alpes.fr.

Depth	CH4 IGE	StDev	tot uncertainty
00.00	1919.7	1.9	15.500
05.00	1934.6	2.1	16.048
07.83	1935.2	2.0	16.081
11.78	1936.2	2.1	15.945
17.91	1931.2	2.5	15.724
23.25	1930.1	2.1	15.539
28.05	1928.8	2.3	15.526
33.01	1922.9	2.1	15.483
38.04	1919.6	1.9	15.462
43.44	1915.3	2.1	15.450
48.20	1910.8	1.9	15.422
51.80	1908.0	2.2	15.430
54.07	1904.4	2.1	15.411
55.75	1902.7	1.7	15.400
57.65	1900.0	2.0	15.407
59.18	1885.9	1.9	15.448
60.86	1855.8	2.3	15.676
62.92	1816.2	1.6	16.605
64.22	1768.6	2.0	18.268
65.84	1657.1	1.8	22.220
66.61	1628.3	1.9	23.951

4 Python code for figure 4

The following code was used to make the plots in fig. 4. Minor adjustments were used to make the remaining figures, with the same underlying idea.

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

path = "C:\\...\\pixelCount\\"
images = ["29.bmp", "71.bmp", "92.bmp", "104.bmp", "107.bmp", "122.bmp",
    "137.bmp", "149.bmp", "164.bmp", "179.bmp"]

for i in images:
    imageIn = i
    imagePath = f"{path}{imageIn}"
    image = cv2.imread(imagePath)
    imgGr = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    pix1 = int(186)
    # for full linescan images:
    #img = imgGr[int((2048/2)-pix1):int((2048/2)+pix1), ...
        ... int(3000):int(3000+(pix1*2))]
    # for images in figure:
    img = imgGr[3000:3372, 800:1172]

    cv2.imshow('image', img)
    cv2.waitKey(1000)
    cv2.destroyAllWindows()

    a = np.empty(51)
    a[:] = np.nan
    b = np.empty(51)
    b[:] = np.nan

    for x in range(0, 255, 5):
        x_write = str(x)
        number_of_pix = np.sum(img <= x)
        if x == 0:
            a[0] = number_of_pix
        else:
            pos = int(x/5)
            number_of_pixLow = np.sum(img <= x-5)
            number_of_pixIn = number_of_pix - number_of_pixLow
            a[pos] = number_of_pixIn
            b[pos] = int(x)
    a[np.isnan(a)] = 0
    b[np.isnan(b)] = 0
    plt.plot(b, a)
    plt.scatter(b, a)

plt.yscale('log')
#plt.ylim((150000))
plt.axis([0, 255, 1, 150000])
plt.xlabel('pixel value []')
plt.ylabel('count []')
plt.title('Pixel value distribution')
plt.show()
```