



THE HONG KONG  
POLYTECHNIC UNIVERSITY  
香港理工大學



DEPARTMENT OF  
LAND SURVEYING AND GEO-INFORMATICS  
土地測量及地理資訊學系

**The Hong Kong Polytechnic University**  
**Department of Land Surveying and Geo-informatics**

**LSGI3322 Satellite Positioning Systems**  
**GPS Positioning Project**  
**(Intermediate 1 – Reading 2 RINEX files)**  
**Subject Lecturer: Prof. George Liu**



**Student Name: Waterdragen**

**Student ID: 24[REDACTED]ID**

**Year 3 student of Land Surveying and Geo-informatics**

## Task

In this exercise I have developed a computer program to open the files for navigation data (.xxn) and observation data (.xxo) and store them in two .csv files.

## Interpretation

The Receiver Independent Exchange Format (RINEX) defines two formats for two files, .xxn file for navigation data, and .xxo file for observation data. The following uses site0900.01n and site0900.01o as an example for our interpretation.

```

1 2 NAVIGATION DATA RINEX VERSION / TYPE
2 CCRINEXN V1.5.9 UX CDDIS 02-APR-00 07:08 PGM / RUN BY / DATE
3 IGS BROADCAST EPHEMERIS FILE COMMENT
4 0.41910-07 0.14900-07 -0.23840-06 -0.59610-07 ION ALPHA
5 0.14950+06 0.00000+00 -0.39320+06 0.39320+06 ION BETA
6 -0.2398081733190-13-0.1396983861920-07 61440 1108 DELTA-UTC: A0,A1,T,W
7 13 LEAP SECONDS
8 END OF HEADER
9
10 2 01 3 31 0 0 0.0-0.3601582720880-03-0.5343281372920-11 0.000000000000+00
11 0.520000000000+00 0.781250000000+01 0.4975564395090-08 0.1872380742160+01
12 0.1620581279830-06 0.2078820555470-01 0.9592622518540-05 0.5153689046860+04
13 0.518400000000+06 -0.2216547727580-06 0.5517310291720+00 0.4414469003680-06
14 0.9335033445960+00 0.1812812500000+03 -0.2056644557920+01 -0.8161054226380-08
15 -0.2632252501030-09 0.000000000000+00 0.110700000000+04 0.000000000000+00
16 0.100000000000+01 0.000000000000+00 -0.1862645149230-08 0.564000000000+03
17 0.511530000000+06 0.000000000000+00 0.000000000000+00 0.000000000000+00
18 3 01 3 31 0 0 0.0-0.1627672463660-04 0.3183231456210-11 0.000000000000+00
19 0.530000000000+02 0.863125000000+02 0.4980207445630-08 0.3494395027580+00
20 0.4433095455170-05 0.2096630050800-02 0.7797032594680-05 0.5153687135700+04
21 0.518400000000+06 0.000000000000+00 0.1631866694900+01 0.59600464477540-07
22 0.9372990537880+00 0.2185312500000+03 0.7149739877990+00 -0.8330346992390-08
23 0.9571827276460-10 0.100000000000+01 0.110700000000+04 0.000000000000+00
24 0.400000000000+01 0.000000000000+00 0.4656612873080-08 0.565000000000+03
25 0.518148000000+06 0.000000000000+00 0.000000000000+00 0.000000000000+00
26 4 01 3 31 0 0 0.0-0.7009222172200-03 0.2501110429880-11 0.000000000000+00
27 0.234000000000+03 -0.581250000000+01 0.4523045545860-08 -0.2112124024710+01
28 -0.2402812242510-06 0.5543017759920-02 0.3607943654060-05 0.5153676870350+04
29 0.518400000000+06 -0.1098960638050-06 0.2746014317970+01 -0.6891787052160-07
30 0.9745701922000+00 0.3216875000000+03 -0.4324080971050+00 -0.8220699567990-08
31 0.2285809498860-10 0.100000000000+01 0.110700000000+04 0.000000000000+00
32 0.200000000000+01 0.000000000000+00 -0.6053596735000-08 0.234000000000+03
33 0.511218000000+06 0.000000000000+00 0.000000000000+00 0.000000000000+00
34 6 01 3 31 0 0 0.0-0.3459863364700-06 -0.1136868377220-12 0.000000000000+00
35 0.570000000000+02 0.106625000000+03 0.4772698802060-08 -0.1200722714170+01
36 0.5621463060380-05 0.6561903865080-02 0.7567927241330-05 0.5153789993290+04
37 0.518400000000+06 0.1527369022370-06 0.1676383757250+01 0.6705522537230-07
38 0.9448496904050+00 0.2204687500000+03 -0.2320140473290+01 -0.8267130073440-08
39 0.1553636143750-09 0.100000000000+01 0.110700000000+04 0.000000000000+00
  
```

According to the RINEX version 2 by Gurtner (1993), and the detailed descriptions by Corcetto et al (1997), the area in orange frame represents each individual navigation data and their descriptions can be interpreted as in the following table:

prn	year, month, day, hour, minute, second (space delimited)	Sv (satellite) clock bias	Sv (satellite) clock drift	Sv (satellite) clock drift rate
IODC (Issue of Data, Ephemeris)	Crs (correction terms to orbital radius)	$\Delta n$ (mean motion difference)	M0 (mean anomaly)	
Cuc (latitude argument correction terms)	Eccentricity	Cus (latitude argument correction terms)	$\sqrt{a}$ (square root of the semi-major axis)	
t0 (or TOE, Time of Ephemeris)	Cic (inclination correction terms)	$\Omega 0$ (right ascension)	Cis (inclination correction terms)	
i0 (inclination)	Crc (orbital radius correction terms)	$\Omega$ (perigee argument)	d $\Omega$ /dt (or omega dot, rate of right ascension)	
di/dt (rate of inclination)	Codes for L2 channels	Week number	L2 P Data flag	
Sv accuracy	Sv health	Tgd	IODC (Issue of Data, clock)	
Transmission time of message	Fit interval	Spare	Spare	

```

1 2.10 OBSERVATION DATA 6 (GPS) RINEX VERSION / TYPE
2 teqc 2000Jul20 WCDA Data collecting20010401 00:12:37UTC PGM / RUN BY / DATE
3 Solaris 2.7|Ultra 2|cc SC5.0|+*|Sparc COMMENT
4 teqc 2000Jul20 WCDA Data collecting20010401 00:12:31UTC COMMENT
5 teqc 2000Jul20 WCDA Data collecting20010331 01:12:17UTC COMMENT
6 site xxxxxx MARKER NAME
7 xxxxxxxx MARKER NUMBER
8 xxxxxxxx OBSERVER / AGENCY
9 2025 AOA BENCHMARK ACT 3.3.32.2N 1k99/07/28 REC # / TYPE / VERS
10 95174 AOAD/M_T EMRA ANT # / TYPE
11 00000001.0000 00000001.0000 0000001.0000 APPROX POSITION XYZ
12 0.1000 0.0000 0.0000 ANTENNA: DELTA H/E/N
13 7 L1 L2 C1 P1 P2 S1 S2 WAVELENGTH FACT L1/2
14 30.0000 # / TYPES OF OBSERV
15 INTERVAL
16 STATION INFORMATION EFFECTIVE FROM: 2000/03/15 18:18:00 COMMENT
17 TO: CURRENT DATE (above) COMMENT
18 BIT 2 OF LLI FLAGS (+4) DATA COLLECTED UNDER 'AS' CONDITION COMMENT
19 L1 PHASE CENTRE 0.110m ABOVE ARP COMMENT
20 L2 PHASE CENTRE 0.128m ABOVE ARP COMMENT
21 where ARP is the Antenna Reference Point for HI measurement COMMENT
22 CONTACT xxxxx@pgc.nrcan.gc.ca FOR ADDITIONAL INFORMATION COMMENT
23 P1 = P1 TurboRogue; = Y1 Benchmark COMMENT
24 L1 = L1(CA) COMMENT
25 P2 = P2 TurboRogue; = Y2 Benchmark COMMENT
26 L2 = L2(P2) TurboRogue; = L2(Y2) Benchmark COMMENT
27 SNR is mapped to RINEX snr flag value [1,4-9] COMMENT
28 SNR: >310 >100 >31.6 >10 >3.2 >0 bad=0 COMMENT
29 L1 & L2: 9 8 7 6 5 4 1 COMMENT
30 2001 3 31 0 0 0.0000000 GPS TIME OF FIRST OBS
31 END OF HEADER
32 01 3 31 0 0 0.0000000 0 106106226176156186 66266286236 3
33 -4571014.571 4 -3561795.634 4 23688534.679 23688535.001 23688542.297
34 37.000 27.000
35 -8749437.103 5 -6817728.119 5 22540847.552 22540847.117 22540852.845
36 66.000 48.000
37 -24594284.015 9 -19164312.035 9 20429273.904 20429273.599 20429274.798
38 225.000 315.000

```

As described in the RINEX specification (1993) and the label, the number in blue frame is the number of observations, and the array of strings in green, delimited by space, represents the types of observations. Here we have 7 observations, which are L1, L2, C1, P1, P2, S1, and S2.

```

33 01 3 31 0 0 0.0000000 0 106106226176156186 66266286236 3
34 -4571014.571 4 -3561795.634 4 23688534.679 23688535.001 23688542.297
35 37.000 27.000
36 -8749437.103 5 -6817728.119 5 22540847.552 22540847.117 22540852.845
37 66.000 48.000
38 -24594284.015 9 -19164312.035 9 20429273.904 20429273.599 20429274.798
39 225.000 315.000
40 -20480724.142 9 -15938945.809 8 20781801.352 20781801.110 20781803.540
41 255.000 209.000
42 -16464808.499 6 -11411535.110 5 21937921.945 21937922.864 21937925.421
43 127.000 74.000
44 -17380788.915 7 -13485772.942 6 21539601.164 21539601.273 21539604.277
45 154.000 105.000
46 -10358499.255 5 -8065294.350 5 22251730.748 22251730.909 22251734.780
47 66.000 58.000
48 -1098017.149 4 -849829.068 4 24134689.870 24134690.816 24134698.149
49 16.000 17.000
50 -16407721.254 8 -12785119.773 7 21490116.573 21490116.135 21490119.224
51 201.000 160.000
52 -938091.866 4 -766562.115 4 24663642.424 24663641.882 24663650.913
53 30.000 15.000
54 01 3 31 0 0 0.0000000 0 106106226176156186 66266286236 3
55 -4468380.809 4 -3481820.827 4 23708065.725 23708065.086 23708073.112
56 59.000 27.000
57 -8728004.983 5 -6801027.809 5 22544925.987 22544925.007 22544931.449
58 67.000 47.000
59 -24611016.429 9 -19177350.349 9 20426089.451 20426089.322 20426090.749
60 222.000 316.000
61 -20533081.150 9 -15999743.386 8 20771898.014 20771897.814 20771900.188
62 275.000 212.000
63 -14735225.122 7 -11480384.220 5 21921108.106 21921108.196 21921111.406

```

As described in the RINEX specification (1993), the first line in orange frame stores the epoch, epoch flag, PRN count, and the PRNs. The subsequent rows in blue frame represent individual observations of each type, in this case, L1, L2, C1, P1, P2, S1, S2 respectively. In this exercise, I am only reading the signal from C1 (Coarse/Acquisition) code, which is modulated on the L1 signal. If a line reaches 80 characters, it goes on to the next line. Here is a table for more detailed interpretation:

year (yy), month, day, hour, minute, second (space delimited)			epoch flag	PRN count and PRNs		
L1 (single digit signal strength, optional)	L2 (single digit signal strength, optional)	C1 (single digit signal strength, optional)	P1 (single digit signal strength, optional)	P2 (single digit signal strength, optional)	S1 (single digit signal strength, optional)	S2 (single digit signal strength, optional)
L1	L2	C1	P1	P2	S1	S2
(Repetition until the row number reaches the PRN number)						

Other notes:

- Epoch flag 0: OK, 1: power failure between previous and current epoch, >1: Event flag.

- Single strength ranges from 1-9, the higher the stronger. 0 or blank means not known or don't care.
- The first number of the PRN string is the PRN count. Subsequent numbers following the PRN count are the PRNs. For example "10G10G22G17G15G18G 6G26G28G23G 3" represents 10 PRNs: 10, 22, 17, 15, 18, 6, 26, 28, 23, 3.

## Computer program

I used Python program for this exercise, and it requires at least version 3.7. The main.py file should be in the same directory as the site0900.01n and site0900.01o.

## Methodology

In my program I make use of the labels and make helper functions to advance the file object to the line as shown below.

```
from dataclasses import dataclass
from os import path

OBS_FILE = "site0900.01o"
NAV_FILE = "site0900.01n"
OBS_CSV_FILE = "site0900-01o.csv"
NAV_CSV_FILE = "site0900-01n.csv"

if not path.exists(OBS_FILE):
    print(f"Please make sure '{OBS_FILE}' is placed in the same directory as main.py, aborting...")
    exit(1)
if not path.exists(NAV_FILE):
    print(f"Please make sure '{NAV_FILE}' is placed in the same directory as main.py, aborting...")
    exit(1)

FIELD1 = slice(3, 22)
FIELD2 = slice(22, 41)
FIELD3 = slice(41, 60)
FIELD4 = slice(60, 79)
```

Importing necessary libraries, defining constants for file names and slices for string slicing when reading navigation files, and ensuring the .xxn and .xxo files exists.

```
def jump_to_suffix_line(file_obj, suffix: str) -> str:
    suffix += "\n"
    line: str
    for line in file_obj:
        if line.endswith(suffix):
            return line.rstrip()
    raise RuntimeError("Cannot find suffix " + suffix)

def jump_to_containing_line(file_obj, s: str) -> str:
    for line in file_obj:
        if s in line:
            return line.rstrip()
    raise RuntimeError("Cannot find text " + s)
```

Helper functions to advance the file stream object to a specific line. `jump\_to\_suffix\_line` can advance the file object to the line ending with the suffix, and similarly, `jump\_to\_containing\_line` can advance it to the line containing the string. This can be useful for locating the lines with string labels.

```
def parse_sat_prn(s: str) -> list[int]:
    # collect the numbers in the string into a list
    # for example: "106106226176156186 66266286236 3" becomes
    # [10, 22, 17, 15, 18, 6, 26, 28, 23, 3]
    # and there are 10 satellites
    num_satellites = int(s[30:32])
    return [int(s[33 + i * 3: 35 + i * 3]) for i in range(num_satellites)]
```

This helper function can parse a satellite prn (Pseudorandom Noise) string into a list of prn integers.

```
def parse_time(s: str, four_digit_year=False) -> tuple[int, int, int, int, int, float]:
    # helper function to parse a space delimited time
    tme = s.split()
    year = int(tme[0])
    if not four_digit_year:
        year += 1900 if year >= 80 else 2000
    return (year, int(tme[1]), int(tme[2]),
            int(tme[3]), int(tme[4]), float(tme[5]))
```

This helper function can parse the date string, delimited by space, into a list of numbers. We need to add 1900 or 2000 to the year if it is only two digits.

```
def parse_sci_float(s: str) -> float:
    return float(s[:15]) * 10 ** int(s[16:])
```

This helper function can parse the scientific notations numbers from the navigation file with string slicing because the length is fixed.

```
@dataclass
class FirstObsTime:
    year: int
    month: int
    day: int
    hour: int
    minute: int
    second: float
```

*Dataclass for storing the first observation time*

```
class KeyValueCSV:
    @classmethod
    def header_row(cls) -> str:
        return ",".join(cls.__annotations__.keys()) + '\n'

    def values_row(self):
        return ",".join(str(i) for i in vars(self).values()) + '\n'
```

This base class can convert class fields into csv headers and value rows. Dataclasses instead of tuples are used because of clarity. The dataclasses for navigation and observation files derive from this base class and the dataclass metaclass generates the constructor method.

```
@dataclass
class ObsData(KeyValueCSV):
    prn: int
    year: int
    month: int
    day: int
    hour: int
    minute: int
    second: float
    epoch_flag: int
    c1: float
```

*Dataclass for storing observation data*

```
@dataclass
class NavData(KeyValueCSV):
    prn: int
    year: int
    month: int
    day: int
    hour: int
    minute: int
    second: float
    sv_clock_bias: float
    sv_clock_drift: float
    sv_clock_drift_rate: float

    iode: float      # Issue of Data, Ephemeris
    crs: float       # correction terms to orbital radius
    delta_n: float   # mean motion difference
    m0: float        # mean anomaly

    cuc: float       # latitude argument correction terms
    ecc: float        # eccentricity
    cus: float       # longitude argument correction terms
    sqrt_a: float     # sqrt of semi-major axis
```

```

t0: float      # or toe, Time of Ephemeris
cic: float      # inclination correction terms
omega_0: float  # right ascension
cis: float      # inclination correction terms

i0: float      # inclination
crc: float      # orbital radius correction terms
omega: float    # perigee argument
d_omega: float  # or "dΩ/dt", "omega dot", rate of right ascension

d_i: float      # or "di/dt", rate of inclination
code_l2: float  # codes for L2 channels
week_no: float
l2_p_data_flag: float

sv_accuracy: float
sv_health: float
tgd: float
iodc: float     # Issue of Data, Clock

transmission_time_of_msg: float
fit_interval: float
spare_1: float
spare_2: float

```

*Dataclass for storing navigation data*

```

@classmethod
def parse(cls, f):
    line = next(f)
    prn = int(line[:2])
    year, month, day, hour, minute, second = parse_time(line[2:22])

    sv_clock_bias = parse_sci_float(line[FIELD2])
    sv_clock_drift = parse_sci_float(line[FIELD3])
    sv_clock_drift_rate = parse_sci_float(line[FIELD4])

    misc = []
    for _ in range(7):
        line = next(f)
        misc.append([
            parse_sci_float(line[FIELD1]),
            parse_sci_float(line[FIELD2]),
            parse_sci_float(line[FIELD3]),
            parse_sci_float(line[FIELD4]),
        ])
    return cls(
        prn, year, month, day, hour, minute, second, sv_clock_bias, sv_clock_drift, sv_clock_drift_rate,
        misc[0][0], misc[0][1], misc[0][2], misc[0][3], # broadcast orbit 1
        misc[1][0], misc[1][1], misc[1][2], misc[1][3], # broadcast orbit 2
        misc[2][0], misc[2][1], misc[2][2], misc[2][3], # broadcast orbit 3
        misc[3][0], misc[3][1], misc[3][2], misc[3][3], # broadcast orbit 4
        misc[4][0], misc[4][1], misc[4][2], misc[4][3], # broadcast orbit 5
        misc[5][0], misc[5][1], misc[5][2], misc[5][3], # broadcast orbit 6
        misc[6][0], misc[6][1], misc[6][2], misc[6][3], # broadcast orbit 7
    )

```

Core implementation for navigation data. It first parses the line containing the time, then iterates the number of broadcast orbits (7), the rest of the matrix corresponds to each of the fields in NavData.

```

def read_obs():
    with open(OBS_FILE) as f, open(OBS_CSV_FILE, 'w') as table:
        table.write(ObsData.header_row())

        text_types_of_satellites = jump_to_suffix_line(f, "# / TYPES OF OBSERV")
        type_obs_count = int(text_types_of_satellites[1:6])
        type_obs_names = [t for t in text_types_of_satellites[6:80].split()]
        c1_index = type_obs_names.index("C1")

        text_time_of_first_obs = jump_to_suffix_line(f, "TIME OF FIRST OBS")
        first_obs_time = FirstObsTime(*parse_time(text_time_of_first_obs, four_digit_year=True))

        print(first_obs_time)

        _ = jump_to_suffix_line(f, "END OF HEADER")

        current_line = f.readline()
        while current_line != "":
            # Check the epoch flag is 0 (OK)
            epoch_flag = current_line[27:30]
            if epoch_flag != "0":
                current_line = f.readline()
                continue

            year, month, day, hour, minute, second = parse_time(current_line[:26])

            sat_prn = parse_sat_prn(current_line)

            for prn in sat_prn:
                lines = ""
                new_line = f.readline()
                lines += new_line

                while len(new_line) == 79:
                    new_line = f.readline()
                    # Add an arbitrary character to make it 80, multiple of 16
                    lines += new_line + "\n"

                obs = [float(lines[i * 16: i * 16 + 14]) for i in range(type_obs_count)]

                obs_data = ObsData(prn, year, month, day, hour, minute, second,
                                   int(epoch_flag),
                                   obs[c1_index])
                table.write(obs_data.values_row())

            current_line = f.readline()

        print(f"Successfully wrote observation data to {OBS_CSV_FILE}")

```

Core implementation for reading observation data. Here are the steps:

1. go to the line that ends with “# / TYPES OF OBSERV”. Parse the type count and type names. Find the index of “C1” of the list
2. go to the line that ends with “TIME OF FIRST OBS”. Parse the first observation time and print it to console.
3. jump to “END OF HEADER”
4. start on a line containing an epoch flag == 0, then parse the time and list of prn. For each prn, read space delimited lines until there are no lines reaching 79 characters. Concatenate the lines and slice them into a list of floats, and index with the c1 index previously found.
5. write to csv then repeat until there are no more lines left.



```

def read_nav():
    with open(NAV_FILE) as f, open(NAV_CSV_FILE, 'w') as table:
        table.write(NavData.header_row())

        _ = jump_to_containing_line(f, "END OF HEADER")
        try:
            while True:
                nav_data = NavData.parse(f)
                table.write(nav_data.values_row())
            except StopIteration:
                pass

    print(f"Successfully wrote navigation data to {NAV_CSV_FILE}")

def main():
    read_nav()
    read_obs()

if __name__ == '__main__':
    main()

```

Controller functions `read_nav` and `main`. In `read_nav`, we write the csv header, then we jump to the line containing the label “END OF HEADER” and for each line parse the line into `NavData` and write the values to the csv until the iterator ends. The main function calls `read_nav` and `read_obs`.

### How to run

use the command ``python3 main.py`` in the directory of `main.py`. Use ``python`` for windows operating system. This Python program requires at least Python 3.7. Make sure that ``site0900.01n`` and ``site0900.01o`` are in the same directory as `main.py`. ``main.py`` file has been attached in the .zip file.

### Expected results

```

PS C:\[redacted]\Desktop\Satellite Positioning> python main.py
Successfully wrote navigation data to site0900-01n.csv
FirstObsTime(year=2001, month=3, day=31, hour=0, minute=0, second=0.0)
Successfully wrote observation data to site0900-01o.csv
PS C:\[redacted]\Desktop\Satellite Positioning>

```

the program should print out success messages for writing the csvs and the first observation time.

### Expected CSV results

File Edit View Insert Format Styles Sheet Data Tools Window Help

LibreOffice Calc

10 pt B I U A

Find

Find All Formatted Display Match Case

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC	DD	DE	DF	DG	DH	DI	DJ	DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK	EL	EM	EN	EO	EP	EQ	ER	ES	ET	EU	EV	EW	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK	FL	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV	FW	FX	FY	FZ	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT	GU	GV	GW	GX	GY	GZ	HA	HB	HC	HD	HE	HF	HG	HH	HI	HJ	HK	HL	HM	HN	HO	HP	HQ	HR	HS	HT	HU	HV	HW	HX	HY	HZ	IA	IB	IC	ID	IE	IF	IG	IH	II	IJ	IK	IL	IM	IN	IO	IP	IQ	IR	IS	IT	IU	IV	IW	IX	IY	IZ	JA	JB	JC	JD	JE	JF	JG	JH	JI	IJ	JK	KL	LM	LN	LO	LP	LQ	LR	LS	LT	LU	LV	LW	LX	LY	LZ	MA	MB	MC	MD	ME	MF	MG	MH	MI	MJ	MK	ML	MM	MN	MO	MP	MQ	MR	MS	MT	MU	MV	MW	MX	MY	MZ	NA	NB	NC	ND	NE	NF	NG	NH	NI	NJ	NK	NL	NM	NN	NO	NP	NQ	NR	NS	NT	NU	NV	NW	NX	NY	NZ	OA	OB	OC	OD	OE	OF	OG	OH	OI	OJ	OK	OL	OM	ON	OO	OP	OQ	OR	OS	OT	OU	OV	OW	OX	OY	OZ	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM	PN	PO	PP	PQ	PR	PS	PT	PU	PV	PW	PX	PY	PZ	QA	QB	QC	QD	QE	QF	QG	QH	QI	QJ	QK	QL	QM	QN	QO	QP	QQ	QR	QS	QT	QU	QV	QW	QX	QY	QZ	RA	RB	RC	RD	RE	RF	RG	RH	RI	RJ	RK	RL	RM	RN	RO	RP	RQ	RR	RS	RT	RU	RV	RW	RX	RY	RZ	SA	SB	SC	SD	SE	SF	SG	SH	SI	SJ	SK	SL	SM	SN	SO	SP	SQ	SR	SS	ST	SU	SV	SW	SX	SY	SZ	TA	TB	TC	TD	TE	TF	TG	TH	TI	TJ	TK	TL	TM	TN	TO	TP	TQ	TR	TS	TU	TV	TW	TX	TY	TZ	UA	UB	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL	UM	UN	UO	UP	UQ	UR	US	UT	UU	UV	UW	UX	UY	UZ	VA	VB	VC	VD	VE	VF	VG	VH	VI	VJ	VK	VL	VM	VN	VO	VP	VQ	VR	VS	VT	VU	VV	VW	VX	VY	VZ	WA	WB	WC	WD	WE	WF	WG	WH	WI	WJ	WK	WL	WM	WN	WO	WP	WQ	WR	WS	WT	WU	WV	WW	WX	WY	WZ	XA	XB	XC	XD	XE	XF	XG	XH	XI	XJ	XK	XL	XM	XN	XO	XP	XQ	XR	XS	XT	XU	XV	XW	XX	XY	XZ	YA	YB	YC	YD	YE	YF	YG	YH	YI	YJ	YK	YL	YM	YN	YO	YP	YQ	YR	YS	YT	YU	YV	YW	YX	YY	YZ	ZA	ZB	ZC	ZD	ZE	ZF	ZG	ZH	ZI	ZJ	ZK	ZL	ZM	ZN	ZO	ZP	ZQ	ZR	ZS	ZT	ZU	ZV	ZW	ZX	ZY	ZZ	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC	DD	DE	DF	DG	DH	DI	DJ	DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK	EL	EM	EN	EO	EP	EQ	ER	ES	ET	EU	EV	EW	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK	FL	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV	FW	FX	FY	FZ	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT	GU	GV	GW	GX	GY	GZ	HA	HB	HC	HD	HE	HF	HG	HH	HI	HJ	HK	HL	HM	HN	HO	HP	HQ	HR	HS	HT	HU	HV	HW	HX	HY	HZ	IA	IB	IC	ID	IE	IF	IG	IH	II	IJ	IK	IL	IM	IN	IO	IP	IQ	IR	IS	IT	IU	IV	IW	IX	IY	IZ	JA	JB	JC	JD	JE	JF	JG	JH	JI	IJ	JK	KL	LM	LN	LO	LP	LQ	LR	LS	LT	LU	LV	LW	LX	LY	LZ	MA	MB	MC	MD	ME	MF	MG	MH	MI	MJ	MK	ML	MM	MN	MO	MP	MQ	MR	MS	MT	MU	MV	MW	MX	MY	MZ	NA	NB	NC	ND	NE	NF	NG	NH	NI	NJ	NK	NL	NM	NN	NO	NP	NQ	NR	NS	NT	NU	NV	NW	NX	NY	NZ	OA	OB	OC	OD	OE	OF	OG	OH	OI	OJ	OK	OL	OM	ON	OO	OP	OQ	OR	OS	OT	OU	OV	OW	OX	OY	OZ	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM	PN	PO	PP	PQ	PR	PS	PT	PU	PV	PW	PX	PY	PZ	QA	QB	QC	QD	QE	QF	QG	QH	QI	QJ	QK	QL	QM	QN	QO	QP	QQ	QR	QS	QT	QU	QV	QW	QX	QY	QZ	RA	RB	RC	RD	RE	RF	RG	RH	RI	RJ	RK	RL	RM	RN	RO	RP	RQ	RR	RS	RT	RU	RV	RW	RX	RY	RZ	SA	SB	SC	SD	SE	SF	SG	SH	SI	SJ	SK	SL	SM	SN	SO	SP	SQ	SR	SS	ST	SU	SV	SW	SX	SY	SZ	TA	TB	TC	TD	TE	TF	TG	TH	TI	TJ	TK	TL	TM	TN	TO	TP	TQ	TR	TS	TU	TV	TW	TX	TY	TZ	UA	UB	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL	UM	UN	UO	UP	UQ	UR	US	UT	UU	UV	UW	UX	UY	UZ	VA	VB	VC	VD	VE	VF	VG	VH	VI	VJ	VK	VL	VM	VN	VO	VP	VQ	VR	VS	VT	VU	VV	VW	VX	VY	VZ	WA	WB	WC	WD	WE	WF	WG	WH	WI	WJ	WK	WL	WM	WN	WO	WP	WQ	WR	WS	WT	WU	WV	WW	WX	WY	WZ	XA	XB	XC	XD	XE	XF	XG	XH	XI	XJ	XK	XL	XM	XN	XO	XP	XQ	XR	XS	XT	XU	XV	XW	XX	XY	XZ	YA	YB	YC	YD	YE	YF	YG	YH	YI	YJ	YK	YL	YM	YN	YO	YP	YQ	YR	YS	YT	YU	YV	YW	YX	YY	YZ	ZA	ZB	ZC	ZD	ZE	ZF	ZG	ZH	ZI	ZJ	ZK	ZL	ZM	ZN	ZO	ZP	ZQ	ZR	ZS	ZT	ZU	ZV	ZW	ZX	ZY	ZZ	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC	DD	DE	DF	DG	DH	DI	DJ	DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK	EL	EM	EN	EO	EP	EQ	ER	ES	ET	EU	EV	EW	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK	FL	FM	FN	FO	FP	FQ	FR	FS	FT	FU	FV	FW	FX	FY	FZ	GA	GB	GC	GD	GE	GF	GG	GH	GI	GJ	GK	GL	GM	GN	GO	GP	GQ	GR	GS	GT	GU	GV	GW	GX	GY	GZ	HA	HB	HC	HD	HE	HF	HG	HH	HI	HJ	HK	HL	HM	HN	HO	HP	HQ	HR	HS	HT	HU	HV	HW	HX	HY	HZ	IA	IB	IC	ID	IE	IF	IG	IH	II	IJ	IK	IL	IM	IN	IO	IP	IQ	IR	IS	IT	IU	IV	IW	IX	IY	IZ	JA	JB	JC	JD	JE	JF	JG	JH	JI	IJ	JK	KL	LM	LN	LO	LP	LQ	LR	LS	LT	LU	LV	LW	LX	LY	LZ	MA	MB	MC	MD	ME	MF	MG	MH	MI	MJ	MK	ML	MM	MN	MO	MP	MQ	MR	MS	MT	MU	MV	MW	MX	MY	MZ	NA	NB	NC	ND	NE	NF	NG	NH	NI	NJ	NK	NL	NM	NN	NO	NP	NQ	NR	NS	NT	NU	NV	NW	NX	NY	NZ	OA	OB	OC	OD	OE	OF	OG	OH	OI	OJ	OK	OL	OM	ON	OO	OP	OQ	OR	OS	OT	OU	OV	OW	OX	OY	OZ	PA	PB	PC	PD	PE	PF	PG	PH	PI	PJ	PK	PL	PM	PN	PO	PP	PQ	PR	PS	PT	PU	PV	PW	PX	PY	PZ	QA	QB	QC	QD	QE	QF	QG	QH	QI	QJ	QK	QL	QM	QN	QO	QP	QQ	QR	QS	QT	QU	QV	QW	QX	QY	QZ	RA	RB	RC	RD	RE	RF	RG	RH	RI	RJ	RK	RL	RM	RN	RO	RP	RQ	RR	RS	RT	RU	RV	RW	RX	RY	RZ	SA	SB	SC	SD	SE	SF	SG	SH	SI	SJ	SK	SL	SM	SN	SO	SP	SQ	SR	SS	ST	SU	SV	SW	SX	SY	SZ	TA	TB	TC	TD	TE	TF	TG	TH	TI	TJ	TK	TL	TM	TN	TO	TP	TQ	TR	TS	TU	TV	TW	TX	TY	TZ	UA	UB	UC	UD	UE	UF	UG	UH	UI	UJ	UK	UL	UM	UN	UO	UP	UQ	UR	US	UT	UU	UV	UW	UX	UY	UZ	VA	VB	VC	VD	VE	VF	VG	VH	VI	VJ	VK	VL	VM	VN	VO	VP	VQ	VR	VS	VT	VU	VV	VW	VX	VY	VZ	WA	WB	WC	WD	WE	WF	WG	WH	WI	WJ	WK	WL	WM	WN	WO	WP	WQ	WR	WS	WT	WU	WV	WW	WX	WY	WZ	XA	XB	XC	XD	XE	XF	XG	XH	XI	XJ	XK	XL	XM	XN	XO	XP	XQ	XR	XS	XT	XU	XV	XW	XX	XY	XZ	YA	YB	YC	YD	YE	YF	YG	YH	YI	YJ	YK	YL	YM	YN	YO	YP	YQ	YR	YS	YT	YU	YV	YW	YX	YY	YZ	ZA	ZB	ZC	ZD	ZE	ZF	ZG	ZH	ZI	ZJ	ZK	ZL	ZM	ZN	ZO	ZP	ZQ	ZR	ZS	ZT	ZU	ZV	ZW	ZX	ZY	ZZ	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX	AY	AZ	BA	BB	BC	BD	BE	BF	BG	BH	BI	BJ	BK	BL	BM	BN	BO	BP	BQ	BR	BS	BT	BU	BV	BW	BX	BY	BZ	CA	CB	CC	CD	CE	CF	CG	CH	CI	CJ	CK	CL	CM	CN	CO	CP	CQ	CR	CS	CT	CU	CV	CW	CX	CY	CZ	DA	DB	DC	DD	DE	DF	DG	DH	DI	DJ	DK	DL	DM	DN	DO	DP	DQ	DR	DS	DT	DU	DV	DW	DX	DY	DZ	EA	EB	EC	ED	EE	EF	EG	EH	EI	EJ	EK	EL	EM	EN	EO	EP	EQ	ER	ES	ET	EU	EV	EW	EX	EY	EZ	FA	FB	FC	FD	FE	FF	FG	FH	FI	FJ	FK	FL
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

## References

- Calais E. *GPS Geodesy – Lab 5 From GPS ephemerides to ECEF satellite positions*. Retrieved from [https://www.geologie.ens.fr/~ecalais/teaching/gps-geodesy/lab\\_5.pdf](https://www.geologie.ens.fr/~ecalais/teaching/gps-geodesy/lab_5.pdf)
- Crocetto N., Gatti M. & Perfetti N. (1997) *The GPS Single Point Positioning: A Data Processing Program for Tutorial Purposes*. ISPRS Commission VI, Working Group 3. 1-5 Retrieved from [https://www.isprs.org/proceedings/XXXII/6-W1/131\\_XXXII-6-W1.pdf](https://www.isprs.org/proceedings/XXXII/6-W1/131_XXXII-6-W1.pdf)
- Gurtner W. & Mader G. (1990) *RINEX: The Receiver Independent Exchange Format Version 2*. CSTG GPS Bulletin. Vol. 3. Retrieved from <https://www.ngs.noaa.gov/CORS/RINEX-2.txt>
- Mussio L., Crippa B. & Vettore A. (1997) *International Society for Photogrammetry and Remote Sensing* ISPRS 32:1-11. Retrieved from [https://www.researchgate.net/profile/Gabriella-Caroti/publication/260934448\\_Kinematic\\_GNSS\\_SurveyExperiences\\_to\\_be\\_Transferred/links/58f759d50f7e9be34b3426db/Kinematic-GNSS-SurveyExperiences-to-be-Transferred.pdf](https://www.researchgate.net/profile/Gabriella-Caroti/publication/260934448_Kinematic_GNSS_SurveyExperiences_to_be_Transferred/links/58f759d50f7e9be34b3426db/Kinematic-GNSS-SurveyExperiences-to-be-Transferred.pdf)