Getting started:

1)Installation

```
>git clone https://github.com/wdklotz/simulinac XXX
```

Use the command above to clone the latest version into the directory XXX.

2)run the sample

Make shure you use pyhton 3

cd to XXX

```
>python simu.py [<input-file>]
```

3) If it works - bingo!

4)Input (*.yml)

Default sample input-file is „XXX/yml/25_09_2017_versuche_70_200MeV.yml“.
This input file is loaded with lots of comments which explain how to set up input for simu.py.
Be aware: its syntax is YAML which is very picky with indentation!
Edit *filepath* at the end of simu.py if you want other default input files.

5)Features of simu.py can be controlled by following definitions:

5.1)flags:

Most of them are commented. If so, they are set to their default values which are shown in brackets {....}. If you want to activate uncomment! Most important are:

sigma: **if** True beam size is calculated from sigma matrix formalism **else** from twiss functions.

express: **if** True quadrupoles are replaced by fast implementations of thin-lense matrices **else** the thin-lens is calculated as triplet product D*KICK*D.

5.2)sections:

Two sections are defined in the sample LE and HE like

```
- [&<section alias><space><name>,&<section alias><space><name>,….]
```

Only HE is used in the sample.

5.3)parameters:

Physics and lattice parameter definitions like:

```
- <name>: &<parameter alias><space><value> | <value> | *<parameter alias>
```

Some parameter names (see the sample input) are recognized by the program and used as starting conditions. They are: *aperture*, *betax_i*, *alfax_i*, *emitx_i*, *betay_i*, *alfay_i*, *emity_i*, *sigmaz_i*, *dp2p_i*, *frequency* and *Tkin*. Others are of free choice to be used in element definitions.

5.4)elements:

Definition of elements

```
1st line        - <ID>: &<element alias>          #any ID you like
2nd line              - type: <CLASS>             #defines the lattice element
attribute 1           - <key>: *<parameter alias> | <value>
:
attribute N           - <key>: *<parameter alias> | <value>
```

All attribute keys with exception *sec*: *<*section alias*> are mandatory and depend on the element CLASS.

5.4.1)Element CLASS definitions:

I	NOP unit
D	Drift
SIXD	Drift with SixTrack mapping
QF	thick F quadrupole with **slices** option
QD	thick D quadrupole with **slices** option
RFG	RF gap with **mapping** options (ref. T3D and A.Shishlo)
RFC	RF cavity as triple product D*RFG*D
GAP	Simple zero length RF-gap w/o (s,dp/p) (ref. Dr.Tiede & T.Wrangler)
M	Marker with action attributes
SD	T3D sector dipole in x-plane
RD	T3D rectangular dipole x-plane
WD	T3D dipole wedge x-plane
QFth	thin F quadrupole
QDth	thin D quadrupole
QFthx	thin express F quadrupole
QDthx	thin express D quadrupole

5.4.2)CLASS „RFG“ has a **mapping** attribute which can be one of „***t3d***“, „***simple***“, „***base***“, „***ttf***“ or „***dyn***“. „*base*“ is a simplified non-linear map assuming constant time-transition factors. „*simple*“ is pyOrbit‘s linear map. „*t3d*“ uses TRACE 3-D matrix maps and gives same results as „*simple*“. „*ttf*“ uses pyOrbit‘s 3 point Transit Time Factors RF gap-model using Ez(r=0) field data calculated with SuperFish. „*dyn*“ uses the DYNAC gap-model focused on a numericla step-by-step method based on the 5-points Bode‘s rule with Picht coordiante transformations. It uses the same externaly provided SuperFish data.

5.4.3)CLASS „RFG“ has also the **SFdata** attribute which has to be set to the name of the file containing the field profile data from SuperFish.

5.4.4)CLASS „QF“ and „QD“ have a **slices** attribute which can take values [1,…,N]. **if** N=1 the thick T3D matrix is used **elif** N>1 the quadrupole is cut into N slices of thin-lens quadrupoles, either as D*KICK*D or „express“.

5.4.5)CLASS „SIXD“ is a **symplectic drift space mapping** using SixTrack canonical coordinates. Slows the calculation sensibly down and is useful to check final results.

5.4.6)CLASS RFC uses RFG with „*t3d*“ mapping as zero length cavity kick.

6)segments:

A segment is defined as:

```
        - <seg name>:              #any name you like
                - *<element alias>
                - *<element alias>
                  :
```

7)lattice:

Finally the lattice is defined as

```
        1st line        - title: <any text>     # must be present as 1st line!
        lines below             - [N, <seg name>,<seg name>,…,<seg name>]
        lines below             - [N, <seg name>,<seg name>,…,<seg name>]
             :
```

Number N expands the segment list in same brackets N times. Many segment lines can be given, for instance one for each section.