# Development of the online simulation tool ASTREG

Grigory Pomogaybo, National Research Nuclear University MEPhI, Russia

August, 2016

Supervisor: Klaus Floettmann

Advisors: Benno Zeitler,

Stephanie Manz

## Abstract

This report presents my work during the DESY Summer Student Programme 2016 in the MPY group. My primary task was to setup an online model for the REGAE accelerator – program ASTREG, which could read machine parameters from the control system and transfers them to the ASTRA. The report contains a description of this program, work of its components and results obtained when ASTREG was used.

## Contents

# 1. Introduction

## 1.1 REGAE accelerator

REGAE the Relativistic Electron Gun for Atomic Exploration is a small electron accelerator build and operated within the framework of the Center for Free-Electron Laser Science CFEL, i.e. in a collaboration of the Max Planck Society, the University of Hamburg and DESY. REGAE provides high quality electron bunches for time resolved diffraction experiments and serves as test bed for accelerator R&D.

*Figure 1.1 Overview of the REGAE accelerator*

REGAE employs a photo cathode RF gun operated at 3 GHz (S-Band) for the production of electrons. The 1½ cell gun cavity, a scaled version of the cavity in operation at the FLASH FEL, accelerates the electrons to energies of up to 5 MeV. A second RF cavity is operated in bunching mode, i.e. the electrons pass the cavity at the zero crossing of the field, such that a correlated energy spread is introduced.

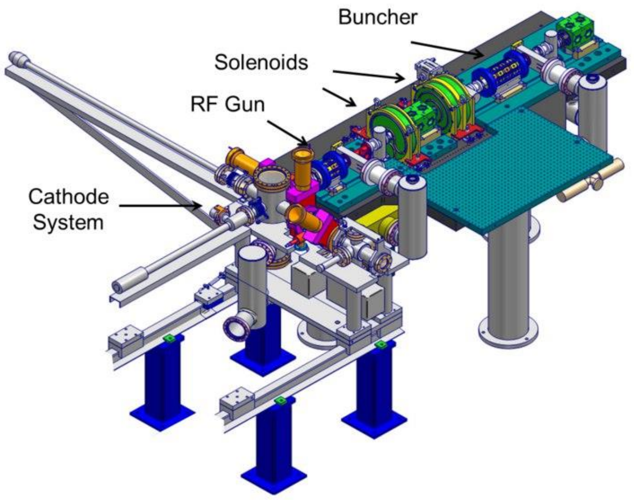



*Figure 1.2 REGAE components*

In the following drift section the bunch length decreases due to the different velocities and reaches a minimum about 4 m downstream of the bunching cavity where the target chamber is located. Extraordinary emittance requirements in the nm range (normalized) and pulse lengths down to a level of ~10 fs require operation at low bunch charges on the sub-pC scale.[1]

On Figure 1.3 we can see the position of all REGAE elements. Subsequently, these positions will be needed to create the input file for ASTRA's calculations. (Basically, ASTREG creates the input file without steerers and dipole magnet, but if it will be nessesary, it can be simply modified.)

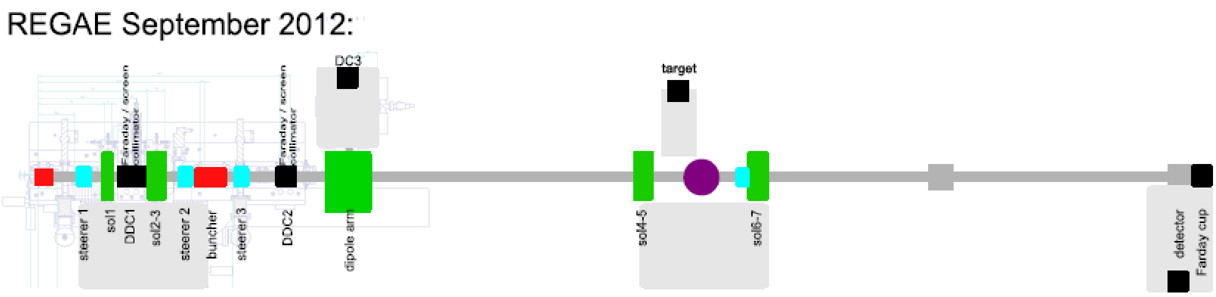



*Figure 1.3 Position of REGAE components*

| Component | Position, mm |
|---|---|
| **Gun, start flange** | -62,20 |
| **Cathode** | 0,00 |
| **Gun, center** | 40,53 |
| **Gun, end flange** | 143,26 |
| **Laser mirror edge** | 286,90 |
| **Laser mirror port, center** | 290,90 |
| **Steerer 1** | 352,00 |
| **Solenoid 1, center** | 550,00 |
| **DDC 1, port 1, center** | 693,00 |
| **DDC 1, port 2, center** | 773,00 |
| **Solenoid 2, center** | 929,50 |
| **Steerer 2** | 1192,00 |
| **Buncher, start flange** | 1200,80 |
| **Buncher, start of first cell (inner wall)** | 1270,00 |
| **Buncher center** | 1360,16 |
| **Buncher, center of first cell** | 1290,00 |
| **Buncher, end flange** | 1519,51 |
| **Steerer 3** | 1654,00 |
| **DDC 2, port 1, center** | 1899,00 |
| **DDC 2, port 2, center** | 1979,00 |
| **Solenoid 3, center** | 5024,90 |
| **Target chamber, center** | 5505,80 |
| **Steerer 4** | 5861,90 |
| **(optional Solenoid 4, center)** | 5986,70 |
| **Detector port 1, center** | 7505,80 |
| **Detector port 2, center** | 9505,80 |
| **Detector port 3, center** | 11505,80 |

## 1.2 A Space Charge Tracking Algorithm (ASTRA)

The Astra (A Space Charge Tracking Algorithm) program package consists of the five parts:

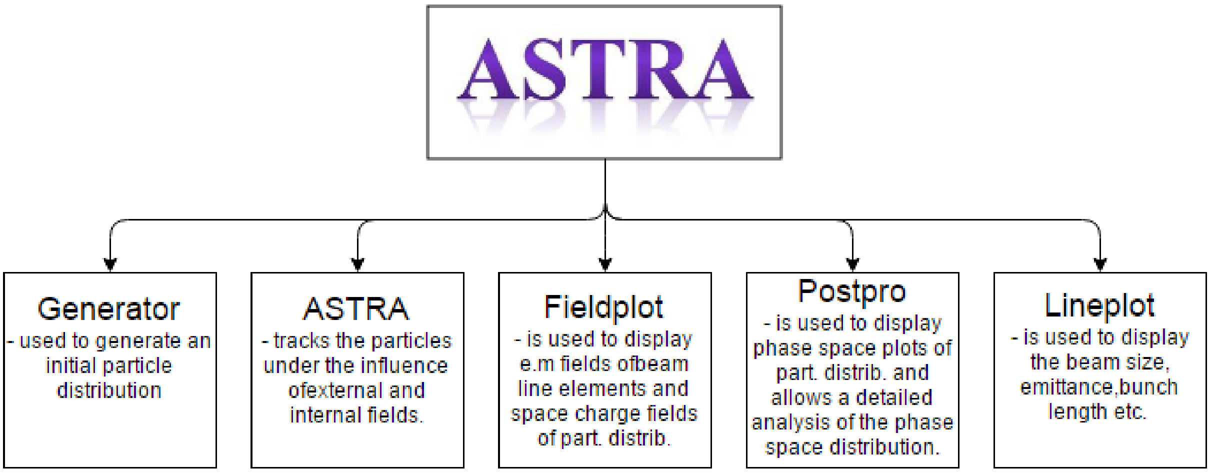



*Figure 1.4 Astra's components*

Astra is written in Fortran 90 and runs on different platforms. The main development platforms are LINUX and Windows. Executables for other platforms are updated less frequently. The menu controlled graphic programs are based on the subroutine package PGPLOT. The input files for the programs generator and Astra are organized in form of Fortran 90 namelists. Each namelist starts with an ampersand (&) followed by the name of the namelist and ends with a slash [2].

**Input namelist for generator**

- INPUT - contains basic instructions for the particle distribution;

**Input namelists for Astra**

- NEWRUN - contains basic instructions for the tracking;
- OUTPUT – contains specifications for the generation of output;
- APERTURE - allows to include apertures and to define material properties for secondary electron emission;
- WAKE - contains the parameters for the wake fields;

- SCAN – contains the parameters for the scanning procedure;
- MODULES - allows to combine elements from other namelists to modules;
- ERROR - allows adding randomly generated errors to element and bunch parameters;
- CHARGE – contains the parameters for the space charge calculation
- CAVITY - allows the user to include, and to some extent to modify arbitrary RF, static electric and magnetic fields;
- SOLENOID - allows to include arbitrary solenoid fields by means of tables, which may be generated by analytical calculations, measurements or numerical codes;
- QUADRUPOLE - allows to include quadrupole fields based on analytical expressions and field profile data;
- DIPOLE - allows to include dipole fields based on analytical expressions

After the input files are created,the user starts the generator and Astra programs. By the end of calculation,the user gets .ref .Xemit .Yemit .Zemit files with data columns. Table 1.1 shows the structure of these files.

*Table 1.1 Results of ASTRA calculation*

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| ref | z, m | t, ns | pz, MeV/c | dE/dz, MeV/m | Larmor angle, rad | $x_{off}$, mm | $y_{off}$, mm | px, eV/c | py, eV/c |
| Xemit | z, m | t, ns | $x_{avr}$, mm | $x_{rms}$, mm | $x'_{rms}$, mrad | $\varepsilon_{x,norm}$, π mrad mm | $x \cdot x'_{avr}$, mrad | | |
| Yemit | z, m | t, ns | $y_{avr}$, mm | $y_{rms}$, mm | $y'_{rms}$, mrad | $\varepsilon_{y,norm}$, π mrad mm | $y \cdot y'_{avr}$, mrad | | |
| Zemit | z, m | t, ns | $E_{kin}$, MeV | $z_{rms}$, mm | $\Delta E_{rms}$, keV | $\varepsilon_{z,norm}$, π keV mm | $z \cdot E'_{avr}$, keV | | |

# 2. ASTREG Manual

## 2.1 Basic concept

The plan for the summer internship was to create a MatLab program which could read machine parameters (magnet currents, rf amplitudes and phases) from the control system and transfers them to ASTRA. After that, Astra should run automatically and the results should be displayed on the control screen. The purpose of the online model is to guide the operator during experiments, help to understand measurements, errors, alignment etc.

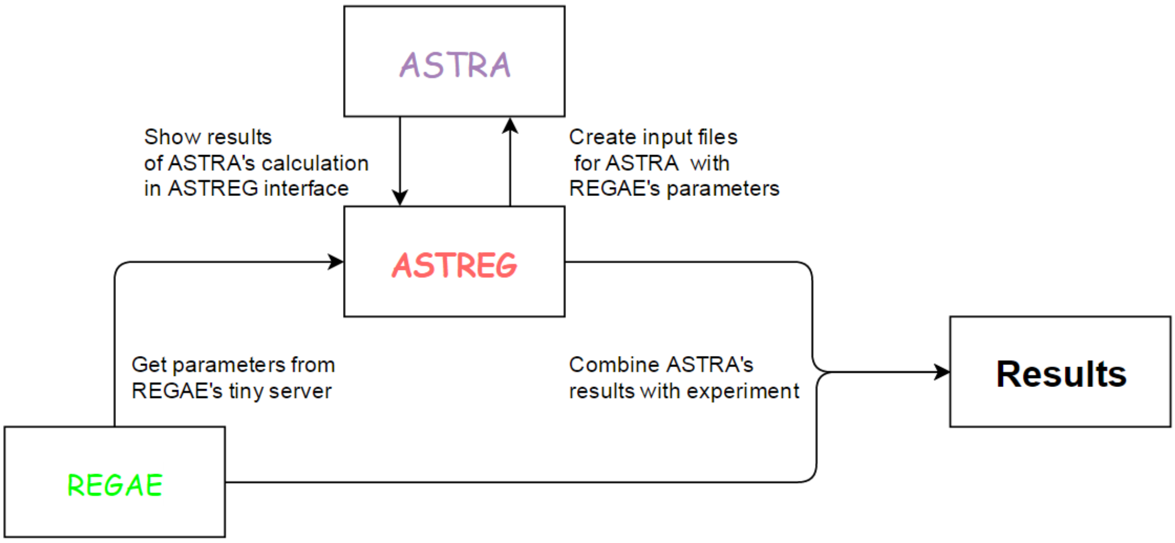



*Figure 2.1 ASTREG application*

For better understanding, ASTREG can be divided into 3 basic parts:

- INPUT – part, which controls incoming data (from REGAE or user) and keeps it in the memory for the next steps.
- RUN/STATUS – part, which controls the creation of .in files for ASTRA. Also, this part starts the ASTRA calculation.
- DISPLAY – part, which contains the graphical interface for ASTRA's results.

## 2.2 INPUT and global parameters

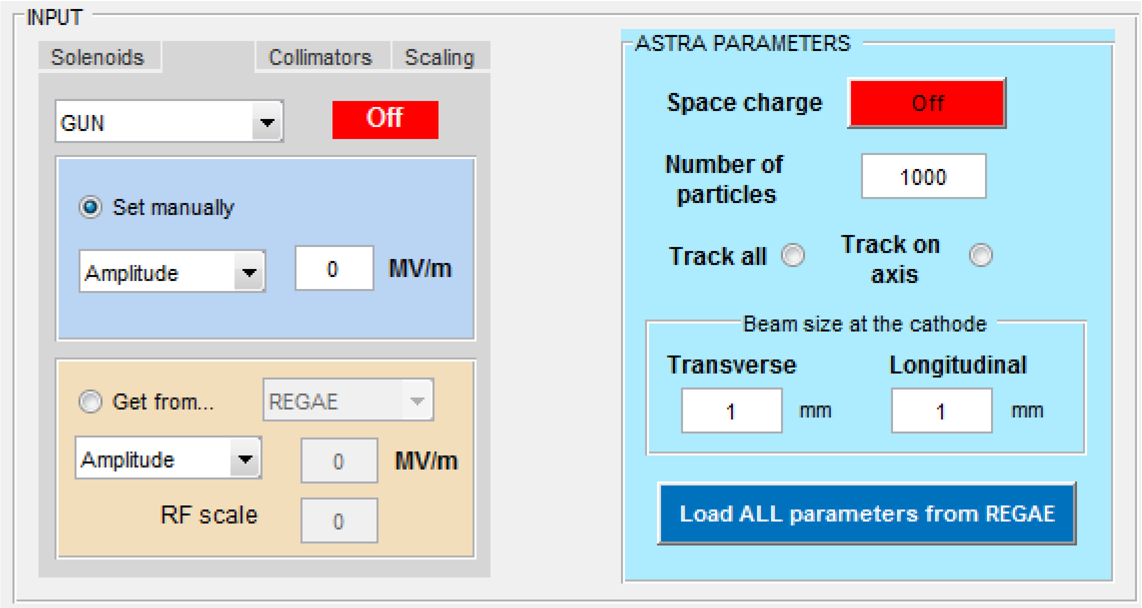



*Figure 2.2 Interface of INPUT part*

Figure 2.2 shows the interface of the INPUT part. It contains one tab panel with REGAE parameters and one panel for ASTRA. The tab panel consists of 4 panels: Solenoids, RF par, Collimators, Scaling. All of these panels are filled with graphical elements, which can be controlled by the user. Before describing the functions, related to these elements, have a look at the global parameters and their purpose:

*Table 2.1 Global parameters*

| Name | | | Description |
|---|---|---|---|
| global ASTREGDATA | .solenoid | .sol1cur<br>.sol23cur<br>.sol45cur<br>.sol67cur | Contains the information about current in solenoids. |
| | | .sol1mag<br>.sol23mag<br>.sol45mag<br>.sol67mag | Contains the information about the max. of the magnetic field. |
| | .rf_par | .gun_ampl<br>.gun_phase | These globals contains GUN’s parameters. |
| | | .buncher_ampl<br>.buncher_phase | Same for buncher |
| | | .gun_scale<br>.buncher_scale | Scaling parameters for buncher and GUN. |
| | .collimators | .col1<br>.col2 | Contains the information about collimator’s holes. |
| | .scaling | .sol1_scal<br>.sol23_scal<br>.sol45_scal<br>.sol67_scal | Contains the information about solenoids scaling parameters. |

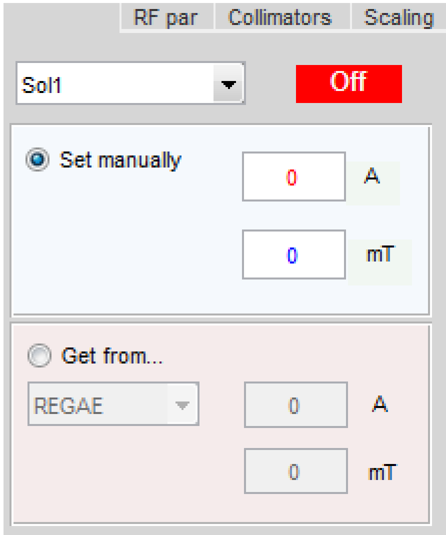
| | | | |
|---|---|---|---|
| | .astrapar | .space_charge | 'boolean' if false, space charge fields are not taken into account. |
| | | .number_of_particles | Contains number of particles to be generated. |
| | | .transverse | rms bunch size in the transverse direction |
| | | .longitudinal | rms bunch size in the longitudinal direction |
| | | .track_all | 'boolean' if false, only the reference particle will be tracked |
| | | .track_on_axis | 'boolean' if true, the reference particle will be tracked only on axis |

| | | |
|---|---|---|
| global ASTREGSERVICE | .astreg_folder | Contains correct path to folder with ASTREG program. |
| | .results_name | Contains correct path to folder with results of ASTRA calculation (or previous results, loaded with button_load_results). |
| | .ampl_or_phase | Indicator, which controls type of incoming value in RF par section (set manually). |
| | .ampl_or_phase_source | Indicator, which controls type of incoming value in RF par section (got from REGAE). |
| | .magnet_id | Indicator, which controls currently operating solenoid, in Solenoid section. |
| | .rf_id | Indicator, which controls currently operating RF component (GUN or Buncher) , in RF par section. |
| | .run_id | Indicator, which controls Astra calculation. If it's equal 0, ASTRA makes a full calculation. Else, it starts the calculation from the place, selected by choose_start element. |
| | .refresh_graf_id | Indicator controls actual type of result, which must be refreshed (actual, when ASTRA is running). |
| | .grid_id | Indicator, which controls grid display. |
| | .collimators_use | Indicator, which controls use collimators or not. |
| | .phase_scan_use | Indicator, which controls use Phase scan or not. |

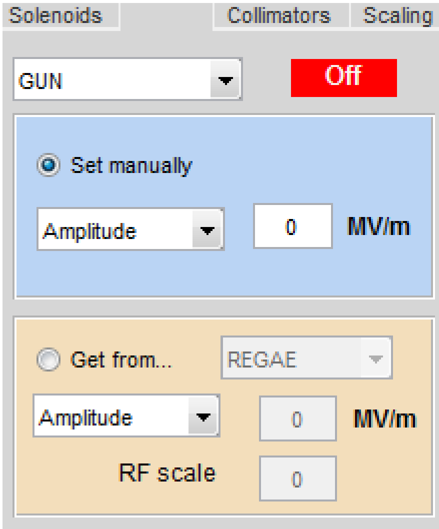
| | | |
|---|---|---|
| global SERVERDATA | .sol1.Strom_Soll | Solenoids actual current, received from tiny server. |
| | .sol23.Strom_Soll | |
| | .sol45.Strom_Soll | |
| | .sol67.Strom_Soll | |
| | .sol1.stat.PS_EIN | Solenoid status, received from tiny server. |
| | .sol23.stat.PS_EIN | |
| | .sol45.stat.PS_EIN | |
| | .sol67.stat.PS_EIN | |
| | .modulator.stat. SysStateRead.value | Modulator status, received from tiny server. |
| | .gun_ampl | GUN's parameters, received from tiny server. |
| | .gun_phase | |
| | .buncher_ampl | Buncher parameters, received from tiny server. |
| | .buncher_phase | |

*Table 2.2 Tab panel elements*

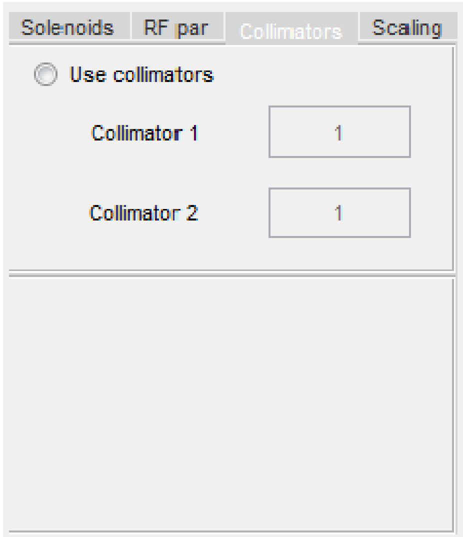
| № | Type | Tag |
|---|---|---|
| **1** | Pop-up menu | choose_magnet |
| **Callback Description** | Loads current and magnetic field from global ASTREGDATA. Also, through ASTREGSERVICE.magnet_id program controls solenoid selection. | |
| **2** | Text | mag_status |
| **Callback Description** | Doesn't have callback func. It shows status of selected magnet (on/off) IMPORTANT – it works only when we get result from the REGAE control system. | |
| **3** | Radiobutton | set_manually |
| **Callback Description** | When it's on, it makes unavailable all interface elements connected with REGAE. | |
| **4** | Edit text | current_value |
| **Callback Description** | User sets current, it will be saved in ASTREGDATA and recalculated into magnetic field (scale parameters from ASTREGDATA also included).After calculation it will be saved in ASTREGDATA.solenoid.sol1mag and sent into magnet_value. | |
| **5** | Edit text | magnet_value |
| **Callback Description** | Reverse to (4) | |
| **6** | Radiobutton | get_magnet_source |
| **Callback Description** | When it's on, it makes unavailable all interface elements connected with manual parameters setup. After that, it connect to matlab tiny server and returns information about selected solenoid (status, current, automatically recalculates current into magnetic field).All parameters will be saved in ASTREGDATA. | |
| **7** | Edit text | current_value_source |
| **Callback Description** | Not usable, but if someone try to change value, it automatically updates it from global parameters. | |
| **8** | Edit text | magnet_value_source |
| **Callback Description** | Same as (7) | |
| **9** | Pop-up menu | mset_source |
| **Callback Description** | (in progress) choose source for ASTREG connection (archive data for example) | |



| № | Type | Tag |
|---|---|---|
| 1 | Pop-up menu | choose_rf |
| Callback Description | Callback function loads amplitude and phase for selected component Also, through ASTREGSERVICE.ampl_or_phase program controls which parameter will be shown in rf_value. | |
| 2 | Text | rf_status |
| Callback Description | Doesn't have callback func. It shows status of selected rf component (on/off) IMPORTANT – it works only when we get result from the REGAE control system. | |
| 3 | Radiobutton | set_manually_rf |
| Callback Description | When it's on, it makes unavailable all interface elements connected with REGAE. | |
| 4 | Pop-up menu | type_rf |
| Callback Description | Loads amplitude or phase of selected REGAE component. Also changes text ( "MV/m" or "deg") | |
| 5 | Edit text | rf_value |
| Callback Description | By the ASTREGSERVICE.rf_id and ASTREGSERVICE.ampl_or_phase program controls type of incoming value and save it in ASTREGDATA. | |
| 6 | Radiobutton | get_rf_source |
| Callback Description | When it's on, it makes unavailable all interface's elements connected with manual parameters setup. After that, it's connect to matlab tiny server and returns information about selected rf component (status, amplitude, phase). | |
| 7 | Pop-up menu | type_rf_source |
| Callback Description | Same as (4) | |
| 8 | Edit text | rf_value_source |
| Callback Description | Not usable, but if someone try to change value, it automatically updates it from global parameters. | |
| 9 | Edit text | rf_scale |
| Callback Description | Contains scaling parameter for RF | |
| 10 | Pop-up menu | rfset_source |
| Callback Description | (in progress) choose source for ASTREG connection (archive data for example) | |



| № | Type | Tag |
|---|---|---|
| **1** | Radiobutton | use_collimators |
| **Callback Description** | When it's on, it makes available Collimator 1 and Collimator 2. When it's off, ASTRA doesn't include APERTURE part in calculation. | |
| **2** | Edit text | col1_value |
| **Callback Description** | User sets radius of collimator's hole, it will be saved in ASTREGDATA. | |
| **3** | Edit text | col2_value |
| **Callback Description** | Same as (2) | |

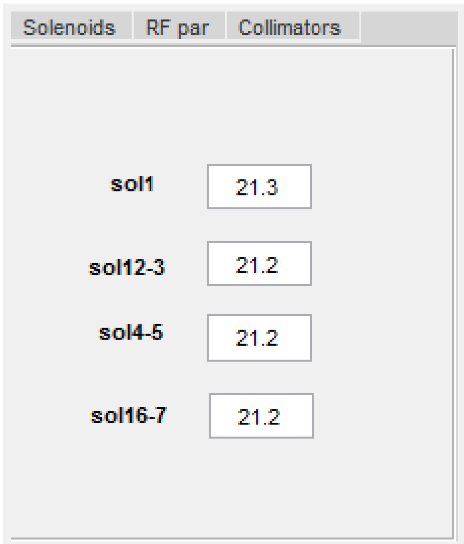



| № | Type | Tag |
|---|---|---|
| **1** | Edit text | sol1_scal |
| **2** | Edit text | sol23_scal |
| **3** | Edit text | sol45_scal |
| **4** | Edit text | sol67_scal |

**Callback Description**

User sets scaling parameter for magnet, it will be saved in ASTREGDATA. It needs to recalculate magnetic field from current:

$$B_{max} = (sol_{scal} * I + 0.3)mT$$

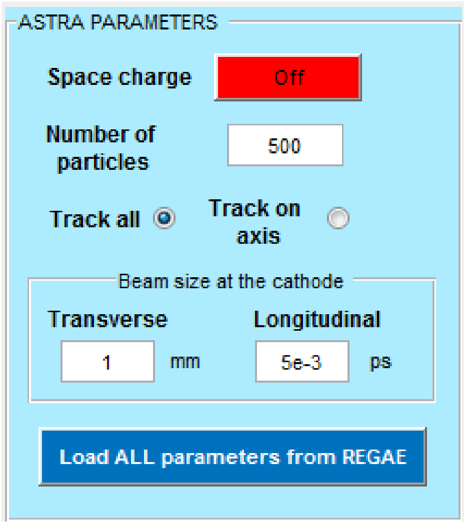



To compare all these panels, ASTREG use Tap_creator function. (It's location ~ \ASTREG\Components\Tab_creator)

Also, INPUT part contains parameters, which need to setup ASTRA:

*Table 2.3 ASTRA PARAMETERS elements*

| № | Type | Tag |
|---|---|---|
| **1** | Button | space_charge |
| **Callback Description** | It controls ASTREGDATA.astrapar.space_charge | |
| **2** | Edit text | number_of_particles |
| **Callback Description** | User sets number of particles, it will be saved in ASTREGDATA. | |
| **3** | Radiobutton | track_all |
| **Callback Description** | It controls ASTREGDATA.astrapar.track_all | |
| **4** | Radiobutton | track_all_on_axis |
| **Callback Description** | It controls ASTREGDATA.astrapar.track_on_axis | |
| **5** | Edit text | transverse |
| **Callback Description** | User sets transverse beam size at the cathode, it will be saved in ASTREGDATA. | |
| **6** | Edit text | longitudinal |
| **Callback Description** | User sets longitudinal beam size at the cathode, it will be saved in ASTREGDATA. | |
| **7** | Button | load_all |
| **Callback Description** | Saves all RF and solenoid parameters in global ASTREGDATA. | |




## 2.3 RUN/STATUS

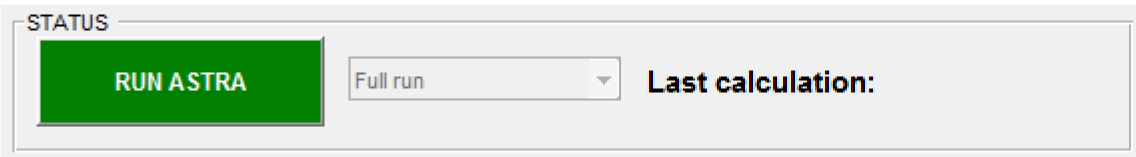
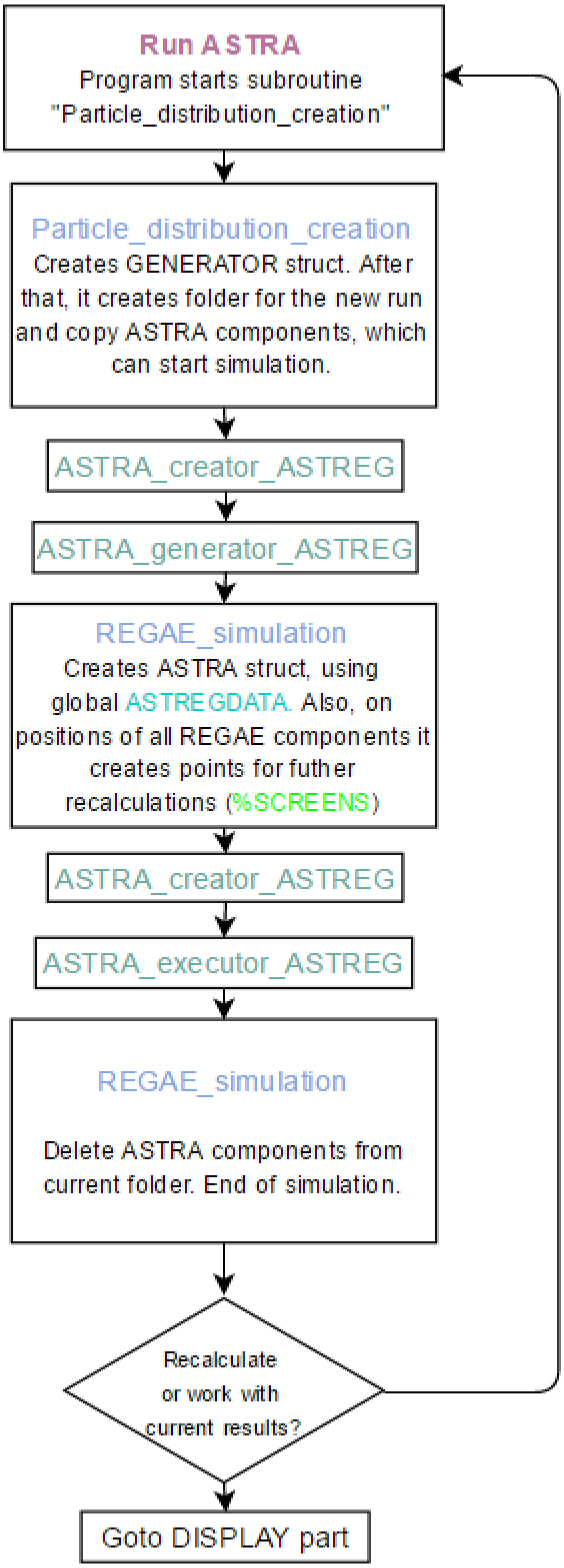
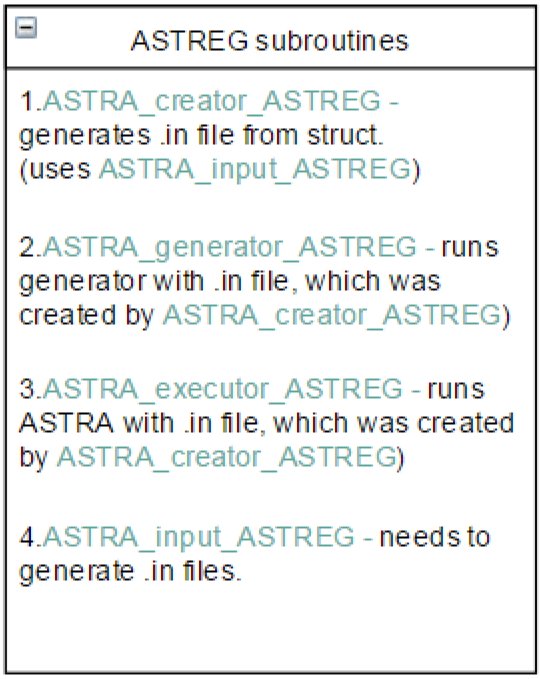



*Figure 2.3 Interface of RUN/STATUS part*

Figure 2.8 shows the interface of RUN/STATUS part. This part controls all manipulations for the ASTRA simulation. From here, user can start a full calculation, or, if a full calculation was done previously, recalculate it, starting with from any REGAE component. Let's have a look at the algorithm of this part:


Run ASTRA
Program starts subroutine "Particle_distribution_creation"
Particle_distribution_creation
Creates GENERATOR struct. After that, it creates folder for the new run and copy ASTRA components, which can start simulation.
ASTRA_creator_ASTREG
ASTRA_generator_ASTREG
REGAE_simulation
Creates ASTRA struct, using global ASTREGDATA. Also, on positions of all REGAE components it creates points for futher recalculations (%SCREENS)
ASTRA_creator_ASTREG
ASTRA_executor_ASTREG
REGAE_simulation
Delete ASTRA components from current folder. End of simulation.
Recalculate or work with current results?
Goto DISPLAY part


ASTREG subroutines
1.ASTRA_creator_ASTREG - generates .in file from struct. (uses ASTRA_input_ASTREG)
2.ASTRA_generator_ASTREG - runs generator with .in file, which was created by ASTRA_creator_ASTREG)
3.ASTRA_executor_ASTREG - runs ASTRA with .in file, which was created by ASTRA_creator_ASTREG)
4.ASTRA_input_ASTREG - needs to generate .in files.

## 2.4 DISPLAY

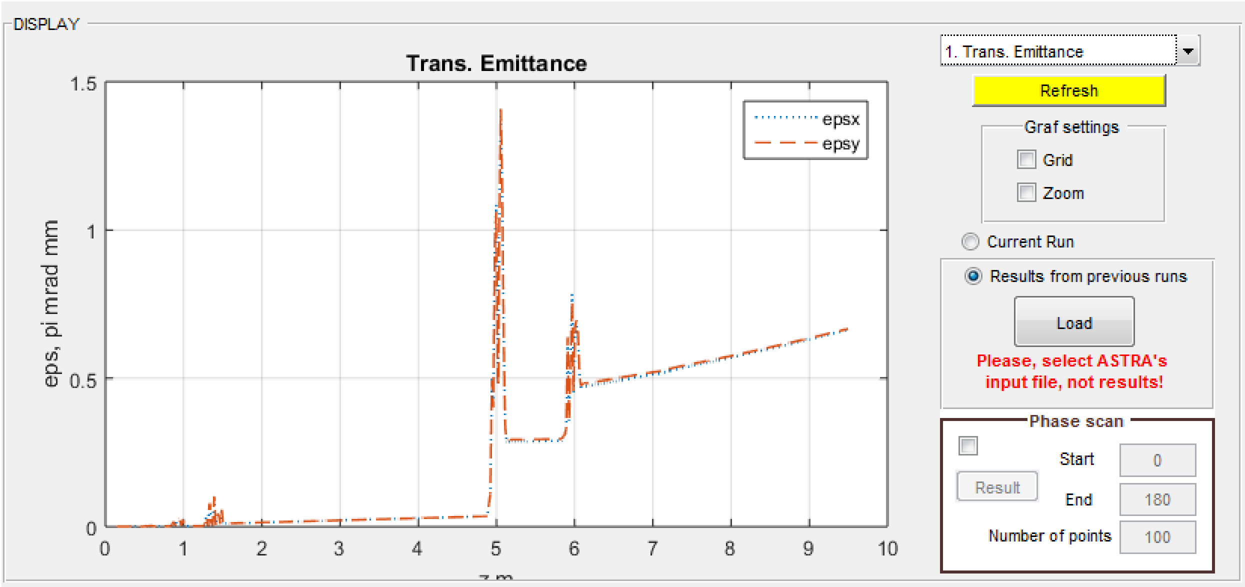



*Figure 2.4 Interface of DISPLAY part*

Figure 2.9 shows the interface of DISPLAY part. This part controls all manipulations with results of ASTRA simulation. Interface contains:

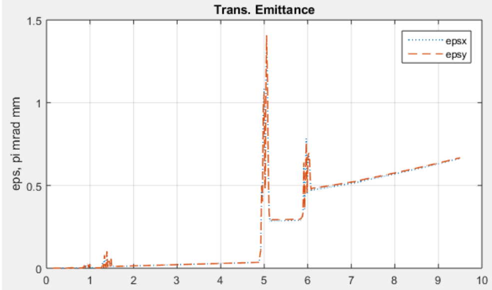
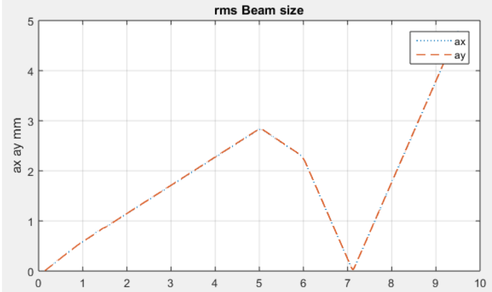
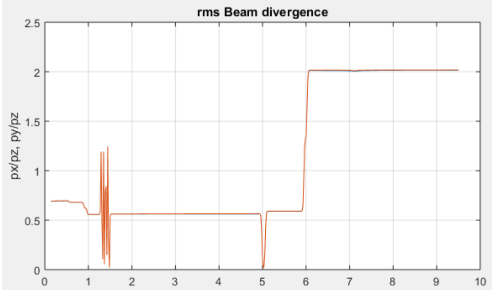
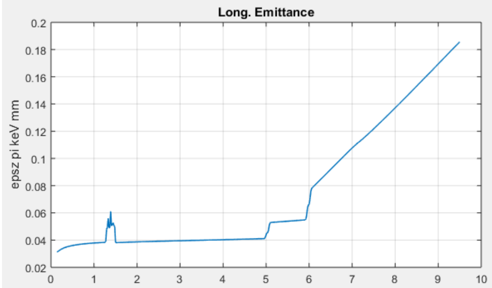
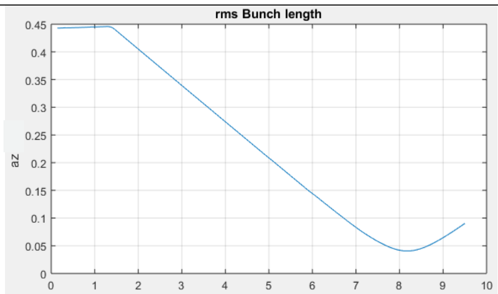
*Table 2.4 Interface description*

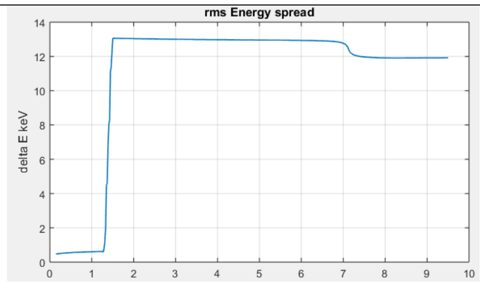
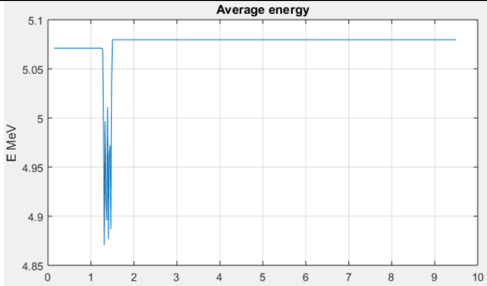
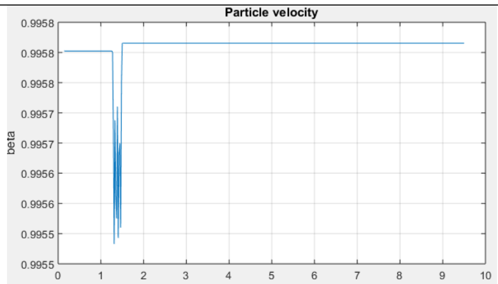
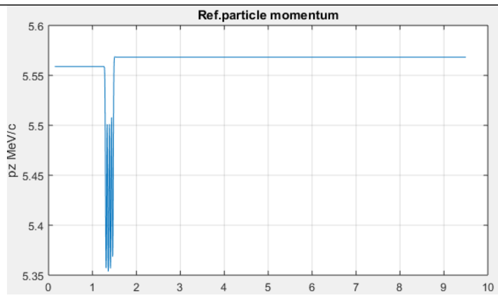
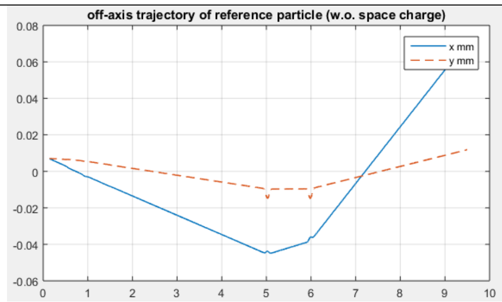
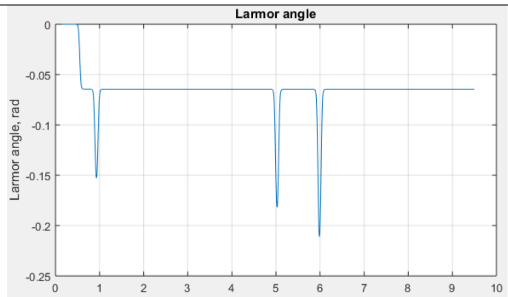
| № | Type | Tag |
|---|---|---|
| **1** | Axes | astra_graf |
| **Callback Description** | Doesn't have callback func. It shows the results, which can be chosen by result_choose. | |
| **2** | Pop-up menu | result_choose |
| **Callback Description** | Callback function execute list of subroutines. When user chooses type of the result, he actually chooses which subroutine must be executed.<br>**List of subroutines:**<br>• alfa_function<br>• average_energy<br>• beta_function<br>• coherence_length<br>• delta_phase_advance<br>• larmor_angle<br>• long_emittance<br>• particle_velocity<br>• phase_advance<br>• ref_particle_trajectory<br>• reference_particle_momentum<br>• rms_beam_divergence<br>• rms_beam_size<br>• rms_bunch_length<br>• rms_energy_spread<br>• trans_emittance | |

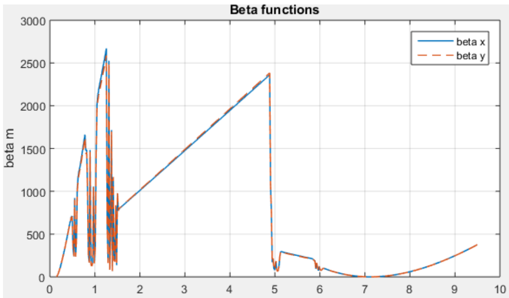
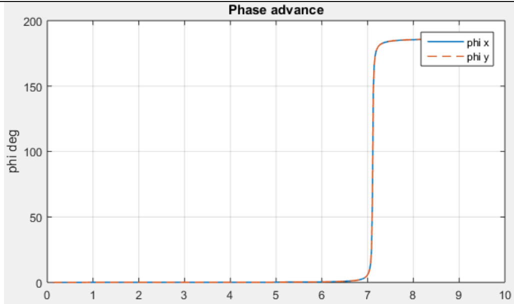
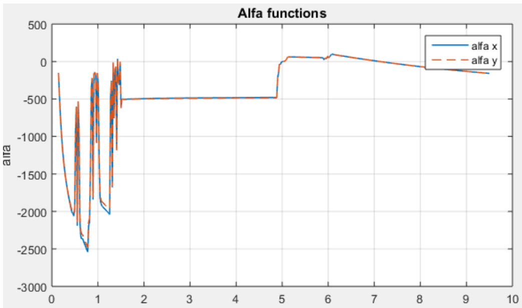
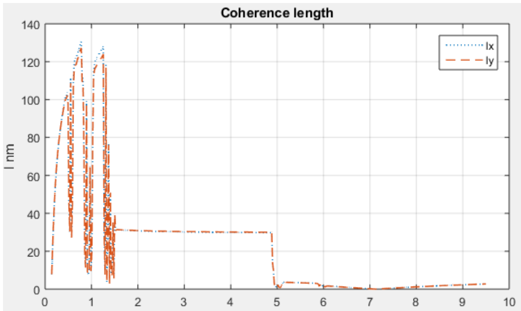
| | | |
|---|---|---|
| **3** | Button | refresh_graf |
| **Callback Description** | (actual, when ASTRA's calculation is running) Updates astra_graf, considering which type of result is set in result_choose. | |
| **4** | Radiobutton | grid_graf |
| **Callback Description** | astra_graf opens with grid | |
| **5** | Radiobutton | zoom |
| **Callback Description** | astra_graf opens with zoom mode | |
| **6** | Radiobutton | current_run |
| **Callback Description** | When it's on, it makes unavailable to load the results from previous run. | |
| **7** | Radiobutton | results_from_previous_runs |
| **Callback Description** | When it's on, it makes available to load the results from previous run. | |
| **8** | Button | button_load_results |
| **Callback Description** | Loads results from previous run. (Important! Callback function doesn´t open results, it just takes filename to set the right path to the results, so user must select ASTRA's input file). | |
| **9** | Radiobutton | use_phase_scan |
| **Callback Description** | Makes available to use phase scan (Important! It should be active before ASTRA calculation) | |
| **10** | Edit text | phase_scan_start |
| **Callback Description** | User sets start point of phase scanning, it will be saved in ASTREGDATA. | |
| **11** | Edit text | phase_scan_end |
| **Callback Description** | User sets end point of phase scanning, it will be saved in ASTREGDATA. | |
| **12** | Edit text | phase_scan_numb |
| **Callback Description** | User sets number of points for simulation, it will be saved in ASTREGDATA. | |
| **13** | Button | phase_scan_result |
| **Callback Description** | Opens phase_scan subroutine. Makes a plot in astra_graf. | |

As we can see, DISPLAY part (particularly part with choosing of the result) is divided into modules (subroutines), so it is very simple to add new modules with result calculation, if it will be needed in future. For now, ASTREG has 16 (+1 phase scanning) subroutines, which take the result of calculation (Table 1.1) and calculates final result for user:

*Table 2.5 Result subroutines description*

| № | Subroutine | Description | Example |
|---|---|---|---|
| 1 | Trans. Emittance | Using .Xemit.001:<br>1 column – z, m<br>6 column - $\varepsilon_{x,norm}$, π mrad mm<br>Using .Yemit.001:<br>6 column – $\varepsilon_{y,norm}$,π mrad mm |  |
| 2 | rms Beam size | Using .Xemit.001:<br>1 column – z, m<br>4 column - $x_{rms}$, mm<br>Using .Yemit.001:<br>6 column – $y_{rms}$, mm |  |
| 3 | rms Beam Divergence | Using .Xemit.001:<br>1 column – z, m<br>5 column - $x'_{rms}$, mrad<br>Using .Yemit.001:<br>5 column – $y'_{rms}$, mrad |  |
| 4 | Long. Emittance | Using .Zemit.001:<br>1 column – z, m<br>6 column - $\varepsilon_{z,norm}$, π keV mm |  |
| 5 | rms Bunch length | Using .Zemit.001:<br>1 column – z, m<br>3 column - $E_{kin}$, MeV<br>4 column - $z_{rms}$, mm<br>$W = E_{kin} + W_{rest}$<br>$p = \sqrt{(W/c)^2 - m^2c^2}$<br>$v = (cp)/\sqrt{c^2m^2 + p^2}$<br>$a_z = z_{rms}/v$ |  |

| | | | |
|---|---|---|---|
| 6 | rms Energy Spread | Using .Zemit.001:<br>1 column – z, m<br>5 column - $\Delta E_{rms}$, keV |  |
| 7 | average Energy | Using .Zemit.001:<br>1 column – z, m<br>3 column - $E_{kin}$, MeV |  |
| 8 | Particle velocity | Using .Zemit.001:<br>1 column – z, m<br>3 column - $E_{kin}$, MeV<br>$W = E_{kin} \times q + W_{rest}$<br>$p = \sqrt{(W/c)^2 - m^2c^2}$<br>$\beta = (p)/\sqrt{c^2m^2 + p^2}$ |  |
| 9 | Reference particle momentum | Using .ref.001:<br>1 column – z, m<br>3 column - pz, MeV/c |  |
| 10 | Ref. particle trajectory | Using .ref.001:<br>1 column – z, m<br>6 column - $x_{off}$, mm<br>7 column - $y_{off}$, mm |  |
| 11 | Larmor angle | Using .ref.001:<br>1 column – z, m<br>5 column - Larmor angle, rad |  |

| | | | |
|---|---|---|---|
| 12 | Beta - function | From (8) we know β factor.<br>Using .Xemit.001:<br>1 column – z, m<br>4 column - $x_{rms}$, mm<br>6 column - $\varepsilon_{x,norm}$, $\pi$ mrad mm<br>Using .Yemit.001:<br>4 column – $y_{rms}$, mm<br>6 column – $\varepsilon_{y,norm}$,$\pi$ mrad mm<br>$\beta_x = x_{rms}^2/\varepsilon_x \times \gamma\beta$<br>$\beta_y = y_{rms}^2/\varepsilon_y \times \gamma\beta$ |  |
| 13 | Phase advance | From (12) we know $\beta_x$, $\beta_y$.<br>Using .Xemit.001:<br>1 column – z, m<br>$\varphi_x = \int \frac{1}{\beta_x}$<br>$\varphi_y = \int \frac{1}{\beta_y}$ |  |
| 14 | Alfa - function | From (12) we know $\beta_x$, $\beta_y$.<br>Using .Xemit.001:<br>1 column – z, m<br>4 column - $x_{rms}$, mm<br>7 column - x·x'$_{avr}$, mrad<br>Using .Yemit.001:<br>4 column – $y_{rms}$, mm<br>7 column – y·y'$_{avr}$, mrad<br>$\alpha_x = -(xx'_{avr}/x_{rms}) \times \beta_x$<br>$\alpha_y = -(yy'_{avr}/y_{rms}) \times \beta_y$ |  |
| 15 | Coherence length | Using .Xemit.001:<br>1 column – z, m<br>4 column - $x_{rms}$, mm<br>6 column - $\varepsilon_{x,norm}$, $\pi$ mrad mm<br>Using .Yemit.001:<br>4 column – $y_{rms}$, mm<br>6 column – $\varepsilon_{y,norm}$,$\pi$ mrad mm<br>$l_{cohx} = 3.8 \times 10^{-4}\, x_{rms}/\varepsilon_x$<br>$l_{cohy} = 3.8 \times 10^{-4}\, y_{rms}/\varepsilon_y$ |  |

## 2.5 Additional information about ASTREG

ASTREG has got many components, which located in different directions. Figure 2.10 shows full map of the program and its components.

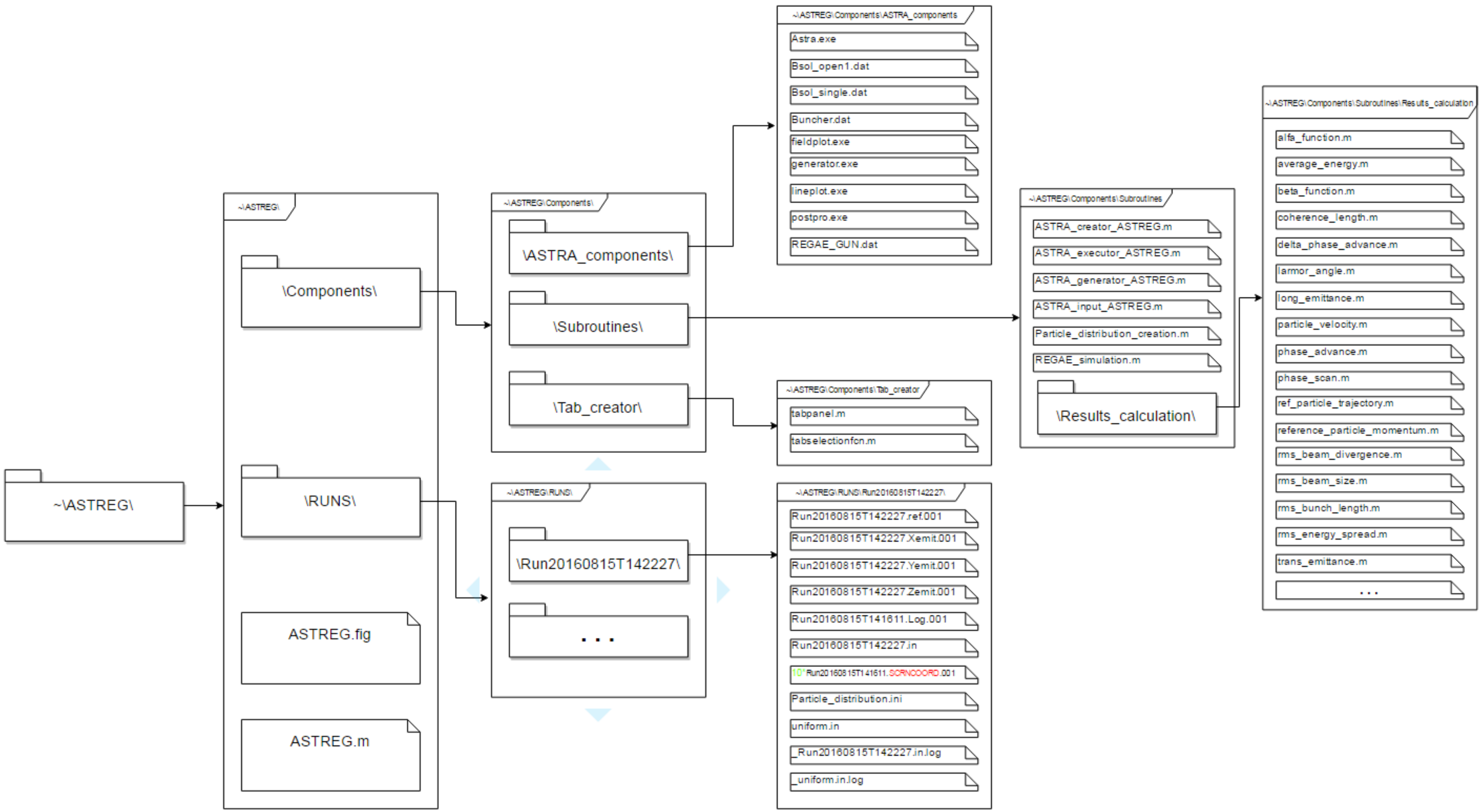



*Figure 2.5 ASTRA's map*

# 3. Experimental part

Astra offers different options to perform parameter scans and optimizations. A simple, predefined scan based on a single particle tracking (the reference particle) is performed by setting 'PHASE_SCAN = True' in NEWRUN. The energy gain as function of the cavity phase is stored as well as the bunch compression factor, i.e. the ratio of the bunch length at the exit of the cavity to the bunch length at the entrance of the cavity. From the derivative of the energy gain w.r.t. the cavity phase a quantity is derived which is proportional to the RF induced energy spread. When the scan for one cavity is finished, the reference particle will be tracked through the cavity on the user-defined phase up to the entrance of the next cavity. Thus, for low energy beams ($\beta < 1$), the result of the scan for downstream cavities depends on the user-defined phase. User defined scans can be performed with the scanning procedure defined in the namelist SCAN.[2]

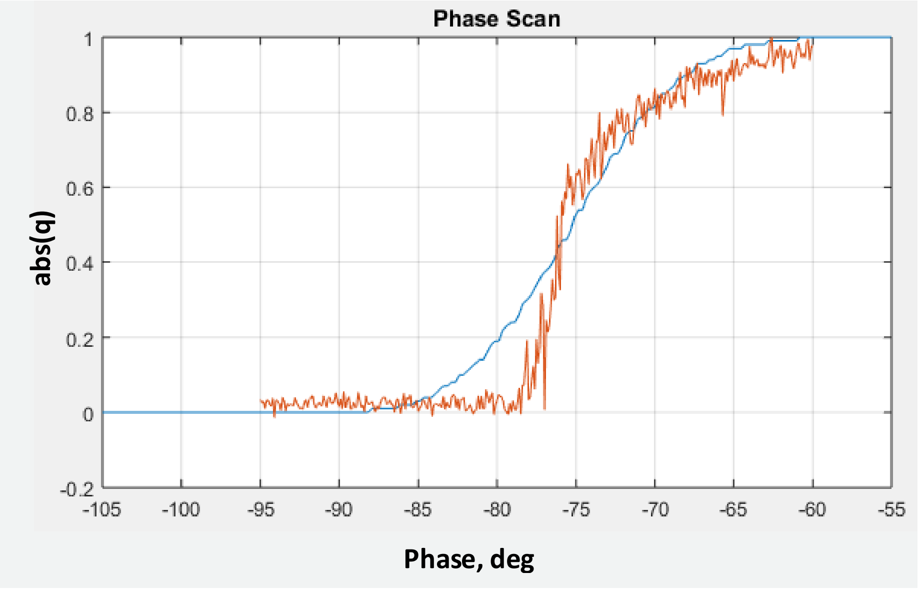



*Figure 3.2 Gun's phase scan. Red - data from REGAE; blue - result of calculation.*

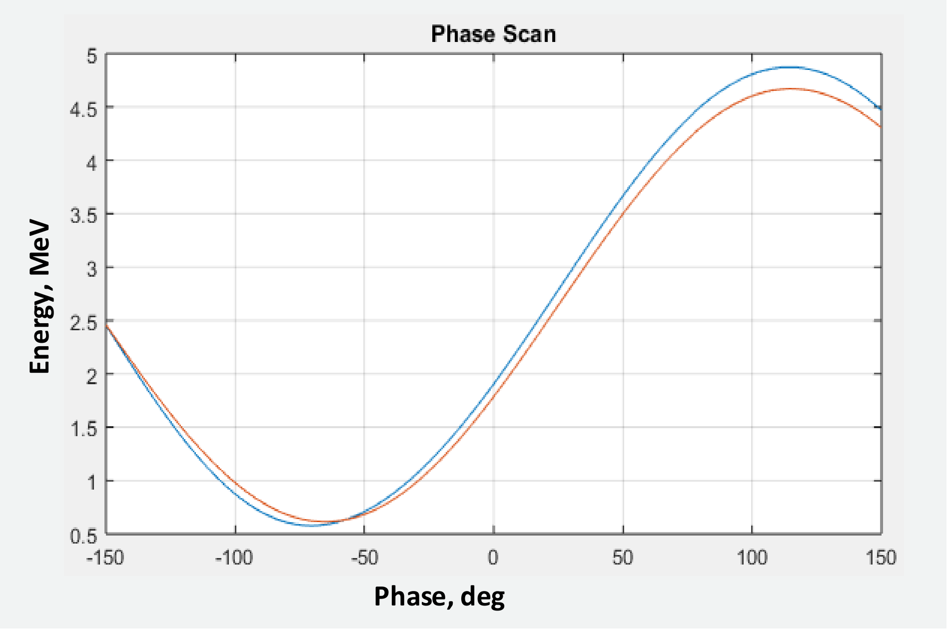



*Figure 3.2 Buncher's phase scan. Red - data from REGAE; blue - result of calculation.*

# 4. Conclusion

By the end of the summer student program, full-operational program ASTREG, was developed. At this moment, the program is able to perform the whole range of tasks, which were set at the beginning of work. Also, a detailed description of how the individual components and the entire program work has been issued. Modular approach to writing the program leaves ample opportunities for a variety of modifications and transformations in the future.

The completion of this work will not be possible without the kind and patient help of my supervisor Klaus, and my advisors Benno and Stephanie. With their help, I've improved my skills at program development and learned so much about basic techniques of accelerator controlling systems. I also want to express my gratitude to all organizers and lecturers of this summer program. Thanks to you, I have learned a lot about modern technologies, which are used in nuclear physics, and consolidated my knowledge of the theory.

# 5. References


[1]. http://regae.desy.de/

[2]. http://www.desy.de/~mpyflo/

[3]. http://adweb.desy.de/mcs/tine/tineMatLabAPI.html

[4]. Klaus Floettmann., *''Rf-induced beam dynamics in rf guns and accelerating cavities''.,* American Physical Society, 2015

[5]. *http://de.mathworks.com/help/matlab/*

[6]. *http://ttfinfo.desy.de/REGAEelog/index.jsp*