**DynacGUI: A Matlab Based Front End for Dynac**

Version 4.1 – 5/29/15

Daniel Alt (alt@nscl.msu.edu)

# Description

This program is a frontend to the accelerator code Dynac, developed by Lapostolle, Valero, and Tanke. It is intended to take separate input files for physical device properties, machine layout, and device settings, generate a properly formatted input deck to the Dynac code, run the code, and allow the user to view the generated plots. It is NOT a beam dynamics code by itself, and requires Dynac to be installed on the machine in question. A copy of the Dynac documentation is also highly recommended.

This frontend is a MATLAB script, and as such also requires MATLAB to be installed on the host machine. While a standalone GUI would be very desirable, and may be implemented in the future, MATLAB currently does not support exporting most of its GUI functionality to an external program. In order to use the fitting tool in DynacGUI, the MATLAB Optimization Toolbox is also required.

This is a work in progress – everything in this documentation is subject to change without notice.

Currently, DynacGUI consists of the following files: (not including user generated input files)

DynacGUI.m – The frontend itself

DynacGUI.fig – The layout file for the frontend

DynacGUI.ini – Initialization file containing default parameters. (The program *should* work if this isn’t present, but don’t depend on it.)

Gendeck.m – The subroutine used to generate the deck

Emitplot.m – Generates plots

DG\_viewresults.m (optional) – Allows viewing of saved results

DynacGUIFit.m (optional) – Allows for basic fitting of beam conditions

DynacGUIFit.fig (optional) – Layout file for the fitting routine

Loadcs.m (optional) – load data from control system output

Dynac can be acquired from the project website at:

http://dynac.web.cern.ch/dynac/dynac.html

# Change Log:

4.0

* Units now displayed on tune edit window
* Modifications to account for changes in Dynac\_r13
* Fitting Tool added
* View Files menu added
* Ability to manually tune bending magnets
* EDFLEC now defaults to requiring 4 values, settable in .ini
* Fixed charges not being output correctly in particle distribution files
* Cosmetic improvements to “Edit Tune Settings” dialog
* Ability to draw dispersion (dp/p vs. x) plots
* Bug fixes and error handling
* Loadcs.m module for importing external tune files.

4.1

* Dispersion plots for multi-charge state beam
* More adjustable parameters in .ini file.
* Updates to manual
* Checking for invalid numbers of sectors in multi-charge state and space charge cases
* Added the ability to rescale bending elements
* Added x, y, time, dE/E and dphi envelopes to z-axis plots
* Added “DynacGUI.ini” to view files menu files menu
* Added support for display of dispersion on z-axis plots when correct Dynac version used.
* Added energy/nucleon to beam data display.
* Bug fixes

Contents

[Description 1](#_Toc418068622)

[Change Log: 2](#_Toc418068623)

[Input files 4](#_Toc418068624)

[Running the frontend 14](#_Toc418068625)

[Tools 18](#_Toc418068626)

[View Files Menu 19](#_Toc418068627)

[DynacGUIFit – Fitting Tool 20](#_Toc418068628)

[Load CS Tune 23](#_Toc418068629)

[Known Issues: 24](#_Toc418068630)

[Appendix A - Sample Files 25](#_Toc418068631)

[Appendix B – Space Charge 27](#_Toc418068632)

# Input files

The program requires three input files – one for the physical machine layout, one for the physical parameters of the various devices and diagnostic devices, and one for the machine settings. It is expected that the first two files will rarely be changed, while the third one will be altered frequently for various device tunes. There is also an .ini file for the program which sets initial parameters.

**Note**: Each file consists of *tab delimited* lists of data. It is very important to use tabs to separate data, or the files will *not* be read correctly.

* 1. **Layout File**

The layout file consists of a list of devices, one to a line, with their associated identifiers. Each device has a different list of parameters. [device name] always refers to the device’s entry in the device types file. [\* Parameter] will always refer to a variable name in the settings file. A tab separated comment is optional at the end of each line

For example, the entry for an electrostatic quadrupole might be:

QUAELEC Quad2 L030V Last Quad of D-line

This would indicate that the physical parameters for the device would be on the “Quad2” line in the device file, and the voltage setting for the device would be found on the “L030V” line in the settings file. “Last Quad of D-line” is a comment.

The currently supported devices are listed below:

* + 1. BMAGNET [device name] [Field Parameter(kG) – Optional]

A bending magnet. If there is no field parameter specified, the field will be set so that the reference particle follows a central trajectory through the magnet. This will also be the case if the parameter is set to zero in the tune settings. (To simulate a bender with zero field, enter an arbitrarily small number, such as .00001 kG)

* + 1. BUNCHER [device name] [harmonic number] [Voltage Parameter] [Phase Parameter]

A sinusoidal buncher. For multiharmonic bunchers, each harmonic should be added as a separate line.

[harmonic number] The multiple of the RF frequency at which the buncher is operating.

* + 1. CAVNUM [device name] [Amplitude Parameter] [Phase Parameter]

An accelerating cavity.

* + 1. DRIFT [length]

A drift.

[length] – Length of the drift, in centimeters.

* + 1. EDFLEC [device name] [Field Parameter(optional)]

An electrostatic deflector. If no parameter is supplied for the field parameter, the voltage will be calculated automatically to keep the reference particle on the desired trajectory. If a parameter name is supplied, this parameter may be used to manually set the deflecting field. (It is still possible to have the field calculated with the parameter present, see below.)

* + 1. NEWF [Frequency (Hz)]

Changes the master frequency of the simulation. Only relevant for RF devices – does not affect DC devices. It is probably good practice to include one of these before the first RF element in each layout file, since the initial RF may be set differently in the tune file.

* + 1. NREF [phase (deg)] [Energy] [flag1] [flag2]

Establishes a new reference particle with a new phase and energy. The flags determine how the new parameters are measured.

Flag1: 0 – Phase is relative to old R.P.

1 – Phase is relative to current c.o.g.

(If REFCOG = 0, these are the same)

Flag2: 0 – [Energy] is change (dW/W) in %

1 – [Energy] is change in MeV

2 – [Energy (MeV)] and [phase (deg)] are both in absolute units.

* + 1. QUADRUPO [device name] [Amplitude Parameter]

A magnetic quadrupole

* + 1. QUAELEC [device name] [Amplitude Parameter]

An electrostatic quadrupole.

* + 1. REFCOG [1 or 0]

Couple or decouple the reference particle from the center of gravity of the particles.

0: Reference particle is defined as the c.o.g.

1: The reference particle and the c.o.g. are independent

2: The reference particle is recalculated at this point at the c.o.g., but then proceeds independently.

* + 1. REJECT [device name]

Reject all particles outside a certain radius or aperture. Note: This rejection threshold is imposed for the remainder of the beamline, or until updated by a subsequent “REJECT” card.

* + 1. RFKICK [device name] [harmonic number] [Amplitude parameter] [Phase Parameter]

(all parameters on one tab-separated line)

A sine wave electric RF kicker. The harmonic number is the multiple of the current reference frequency at which the device operates.

* + 1. RFQPTQ [device name] [Amplitude Parameter] [Phase Parameter]

An RFQ defined from a PARMTEQ simulation.

* + 1. SLIT [Horizontal Parameter] [Vertical Parameter]

This entry, which is a combination of several Dynac commands, adds a horizontal, vertical, or four-jaw slit.

* + 1. STEER [device name] [Field Parameter]

Either an electrostatic or magnetic steerer. Note that this is a zero length thin lens element that produces an instantaneous bend in the beam.

* + 1. STOP

This command causes the frontend to stop processing the layout file at this point. It is useful for temporarily commenting out part of a beamline to save computation time.

* + 1. SOLENO [device name] [Amplitude Parameter]

A solenoid magnet.

* + 1. ZROT [rotation angle (degrees)]

A rotation of the beamline about the z axis. Used for vertical or left turning bends.

In addition, the following output graphs can be specified in the layout file. It is important to put emittance plots at the point in the file where you wish them to be generated. Envelope plots can be specified anywhere in the file \*after\* the end of the envelope. (If you put an envelope plot defined from 0-10 meters at the 5m point in the file, the second 5 meters of the plot will not be plotted.) It is probably simplest to put all the envelope plots at the end of the file, but this is not required.

The plot types are defined in the “device types” file. Plot names may contain spaces, but remember to continue to separate them from the previous parameter by a *tab.*

1. EMIT (no parameters)

This card does not create a plot, but forces Dynac to report emittance and other data at this point in the beamline in the “dynac.short” output file.

1. EMITGR [plot type] [plot name]

An emittance graph. This will produce four plots – x/x’, y/y’, x/y, and dt/dE. When used, DynacGUI will also automatically add an EMITL card to the deck to force a labelled dump of beam data to the “dynac.short” output file at the same point.

1. EMITL [label]

This is the same as the EMIT card, but includes a label for easier identification later.

1. ENVEL [plot type] [plot name]

An envelope plot. This will produce three separate graphs – one of the x and y envelopes along the line, one of dE along the line, and one of dPhi.

1. WRBEAM [file name]

This will write a particle distribution in Dynac format (see the manual) at the specified point on the beamline. By default it uses flags 1 2, for absolute energy in MeV and tabulation of particle charge.

Any lines in the layout file preceded by a “;’ are treated as comments.

* 1. **Device Types File**

This file specifies the physical parameters for each device type, such as length and aperture radius. It is also used to specify dimensions for plots.

In each case, the first entry on the line is a [device type] which should match the [device type] for the selected device in the layout file. As before, each parameter should be on the same line, and separated by tabs. A tab-separated comment is allowed at the end of each line.

Example: (magnetic quadrupole)

Quad2 15 3.75 L-Line extension quad

This would indicate that the device type “Quad2” is 15 cm long, 3.75 cm in radius, and is identified by the comment “L-Line extension quad”.

The parameters for each currently supported device type are:

* + 1. Bending Magnet (BENMAG)

(Remember: all of these parameters should be on the same line)

Device Name – should match the layout file

Bend Angle (degrees) – the bending angle of the magnet

Bend Radius (cm) – the bending radius of the magnet

Entry Face Angle (deg) – the angle of the entry face of the magnet.

Entry Face Curvature (cm) – radius of curvature of the entry face.

Entry ½ Aperture (cm) – the half aperture of the entry face. Set to 0 to neglect fringing effects.

Exit Face Angle (deg)

Exit Face Curvature (cm) – same as for entrance face

Exit ½ Aperture (cm)

The sign conventions for the entry and exit faces are the standard ones from TRANSPORT and other beam codes. (See the Dynac documentation for an illustration.) The number of sectors to be used for calculating this and the electrostatic bender are specified in the .ini file. (below)

* + 1. Buncher (BUNCHER)

[Device Name] [Aperture Radius (cm)]

* + 1. Cavity (CAVNUM)

[Device Name] [Field File]

Field File should be the name of the file containing the field map for the cavity.

* + 1. Electrostatic deflector (EDFLEC)

(Remember – all entries on one tab-separated line)

Device Name – should match layout file

Radius (cm) – bending radius of the deflector

Angle (degrees) – bending angle of the deflector

Vertical Curvature (cm) – radius of vertical curvature. Set to a large number for a cylindrical deflector.

The number of sectors to be used for calculating this and the magnetic bender are specified in the .ini file. (see below)

* + 1. Magnetic quadrupole (QUADRUPO)

[Device Name] [Length (cm)] [Aperture Radius (cm)]

* + 1. Electrostatic quadrupole (QUAELEC)

[Device Name] [Length (cm)] [Aperture Radius (cm)]

* + 1. Rejection threshold (REJECT)

(All parameters on one tab separated line)

[Device name] [Energy ½ width (MeV)] [Phase ½ width (deg)]

[x 1/2 Width (cm)] [y ½ width (cm)] [Radius ½ width (cm)]

* + 1. RF Kicker (RFKICK)

[Device Name] [Kicker direction] [Electrode Length (cm)] [Electrode Gap (cm)]

Kicker Direction:

0 – Horziontal

1 – Vertical

Note that this is a zero length element – the electrode length is only used to calculate the deflection angle.

* + 1. RFQ (RFQPTQ)

(All parameters on one tab separated line)

[Device Name] [File Name] [Number of Cells]

[Design Input Energy (MeV/u)] [Design Frequency (MHz)]

File Name should be a Dynac-formatted RFQ file.

* + 1. Solenoid (SOLENO)

[Device Name] [Length (cm)]

* + 1. Steerer (STEER)

[Device Name] [Steerer Type] [Electrode Length(cm)] [Electrode Gap(cm)]

Steerer Type:

0 – Horizontal Magnetic

1 – Vertical Magnetic

2 – Horizontal Electric

3 – Vertical Electric

Only specify electrode length and electrode gap parameters for electrostatic steerers. Any values entered will be ignored if “type” is 0 or 1. Note that the length is used only for calculations of bending amount – this is a zero length element.

* + 1. Emittance Plot (EMITGR)

(all parameters on one tab separated line)

Plot Type – Should match the type from the layout file

RMS – the RMS at which a circle will be drawn on each graph

x/y limit (cm) – The limit for the x and y graph axes

x’/y’ limit (mrad) – The limit for the x’ and y’ graph axes

dz limit (degrees) – The limit for the dz axis

dE limit (dE/E) – The limit for the dE axis

* + 1. Envelope Plot (ENVEL)

(all parameters on one tab separated line)

Plot Type – Should match type from layout file

RMS – the RMS of the envelope plotted

Start Point (m) – The start of the envelope plot

End Point (m) – The end of the envelope plot. Set to large number for whole line. (Will not plot past end of line)

(NOTE: Not in cm.)

* 1. **Settings File**

The settings file lists the tunable parameters that are expected to change frequently. It also contains the global parameters such as charge state for each tune. Each line should consist of a variable name and a tab or space separatedvalue. A tab or space separated comment is allowed after the value.

Emittances and energy spread are given at the 4RMS level.

* + 1. **Required parameters:**

The following parameters are required, but may be provided in any order:

A Atomic Mass [amu]

Q Charge state [charge units]

RF RF frequency

Npart Number of particles

Energy Total energy of reference particle [MeV]

Alphax, Betax [mm/mrad], Epsx [mm.mrad] Twiss Parameters – x

Alphay, Betay [mm/mrad], Epsy [mm.mrad] Twiss Parameters – y

* + 1. **Longitudinal parameters**

Longitudinal parameters of the initial beam may be specified either as a DC beam (default) or by using Twiss parameters for the z dimensions of the beam. To choose the type of distribution, set the ZLaw parameter as detailed in the Dynac documentation:

ZLaw 1 Hit or miss Monte Carlo within real space

ZLaw 2 Hit or miss Monte Carlo with a 6D ellipsoid

ZLaw 3 Quasi-uniform distribution

ZLaw 4 As with ZLaw=2, but within a Gaussian distribution

ZLaw 5 Uniform distribution in phase, energy, with spread set by Deltae

(If no ZLaw is specified, this will be used)

Either the z Twiss parameters or the energy width must be defined.

Alphaz, Betaz [deg/keV], Emitz [keV.deg] Twiss Parameters - z

Deltae Energy ½ width of particle distribution [MeV]

* + 1. **Multiple Charge State Beams**

Dynac can be used to simulate beams of multiple charge states. If a particle distribution file is used to specify the locations of the input particles, then the charges states will be read from the file with no additional commands needed.

For automatically generated particle distributions, or input distributions without charge states saved, the following lines should be added to the settings file to generate a multi-charge state distribution randomly:

Nstates Number of charge states

cs*x* charge state for state *x*

cspcent*x* percentage of particles with charge state *x*

cseoff*x* energy offset in MeV for charge state *x*

Note that Dynac will return an error if the percentages do not add to 100%. While Dynac supports reading in a list of charge states from an external file, DynacGUI does not support this functionality at this time, except in the case where the entire distribution is read.

Multiple charge state beams require that magnetic and electrostatic benders use more than 1 sector for computation. If only one sector is specified in the .ini file, the value will be changed to 2.

Example:

Nstates 3 (Three charge states)

cs1 24 (Charge states are +24, +25, and +26)

cs2 25

cs3 26

cspcent1 20 (20% of the particles are at +24)

cspcent2 60 (60% of the particles are at +25)

cspcent3 20 (20% of the particles are at +26)

cseoff1 -.01 (The +24 particles are offset by -0.01 MeV)

cseoff2 0 (The +25 particles are at the reference energy)

cseoff3 +.01 (The +26 particles are offset by +0.01 MeV)

* + 1. **ReA specific:**

ZOffset (Only used for ReA3) Distance (m) from L016 to the start of the plot. Used to place diagnostic box guide lines. Only used if “ReABoxes” option is added to dynacgui.ini file.

* + 1. **Device parameters**

The remaining parameters must match the parameters needed for the layout file. Each tunable variable in the layout file must have a corresponding variable on the list. For example, if there is a quadrupole with an amplitude parameter named “L030V”, there must be an entry for the amplitude of the quadrupole field like this:

L030V 2.348

* + 1. **Tune Settings Example:**

A 4

Q 1

RF 80.5E6

Npart 1000

Alphax 0.

Betax .4

Epsx 10.

Alphay 0.

Betay .4

Epsy 10.

Energy .048

ZLaw 5

Deltae .0001097

ZOffset 10.

L030V 4.0

MHB1V .0007

MHB1P -90

L072XG 1000.

L072YG 1

|  |  |
| --- | --- |
| Supported Device Type | Variable(s) |
| Bending Magnet | Bending Field (kG) \*Optional\* |
| Buncher | Amplitude (MV)  Phase (deg) |
| Cavity | Amplitude (%)  Phase (deg) |
| Electrostatic Deflector | Deflecting Field (kV/cm)  Set to ‘-1’ for nominal field |
| Quadrupole (Electric) | Amplitude (kV) |
| Quadrupole (Magnetic) | Amplitude (kG) |
| RF Kicker | Plate Voltage (kV)  Phase (deg) |
| RFQ | Amplitude (%)  Phase (deg) |
| Solenoid | Amplitude (kG) |
| Slit | Horizontal Slit Width [cm]  Vertical Slit Height [cm]  Both numbers are *required* – set unneeded ones large. Note that these are FULL widths, not half widths. |
| Steerer | Electrostatic:  Plate voltage (kV)  Magnetic:  Integrated Field (T.m) |

* 1. **DynacGUI.INI file**

This file contains initial settings for various program parameters. Just as with the other settings file, the format is the setting name and then the value *separated by a tab*. If the .ini file is missing, or does not contain all the values, the program will use internal defaults, which may be unpredictable. Any line not matching a supported setting name or preceded by a semicolon (;) will be treated as a comment. Note that values are case sensitive.

Note that these values are read when DynacGUI is first launched. If they are changed after DynacGUI is running, the values will not be updated until the frontend is closed an relaunched.

Supported values:

File Locations:

Layout default layout filename

Devices default device filename

Tune default tune settings filename

Executable location of Dynac executable. This is unnecessary if Dynac is on your path with the default name of “dynacv6\_0.exe”.

Executable2 and Executable3– locations of a second and third Dynac executable. If this is present, a menu will appear to allow you to select which executable to use. (Potentially useful for comparing Dynac versions.)

Beamline parameters:

RFQreject Fractional energy deviation (ΔE/<E>) rejected after an RFQ. This is the deviation from the average particle energy *including* the unaccelerated particles. (default = 0.5)

Esectors Number of sectors used to calculate electrostatic benders (default = 10)

Bsectors Number of sectors used to calculate magnetic benders. (default = 10)

Edflec Number of parameters accepted by the “EDFLEC” card in Dynac.

3 – Dynac r12 and below.

4 – Dynac r13 and up. (default)

**Note:** While the deck will be generated successfully if this setting is incorrect or absent, a runtime error will be produced if Dynac attempts to execute a card with an incorrect number of parameters.

Other Parameters:

Mingw 1 = Dynac was compiled using the Mingw Fortran compiler

0 = compiled with a different compiler

(this option is ignored on Macs)

ReABoxes 1 = Hard coded device locations for ReA3 boxes can be shown

0 or not present (default) = ReA3 boxes not shown

This option is deprecated, as it is better to use the plot labels.

ReAImport 1 = The “Import ReA Tune” button is visible

0 (default) = Button is not visible

# Running the frontend

1. **Starting the frontend**

From within MATLAB, select and run “DynacGUI.m”. Make sure that all of the following files are present:

DynacGUI.m – The frontend itself

DynacGUI.ini – Initialization file containing default parameters. (The program *should* work if this isn’t present, but don’t depend on it.)

Gendeck.m – The subroutine used to generate the deck

Emitplot.m – Generates plots

Layout file – File describing the machine layout

Device file – File describing the machine devices

Tune Settings file – File containing the machine settings

Field file(s) (optional) – File(s) describing cavity field

RFQ file (optional) – File describing RFQ field

Particle distribution file (optional) – File containing initial particle distribution

DG\_viewresults.m (optional) – Allows viewing of saved results

1. **Select files**

After the input files are setup, run the frontend. Select the layout file, device file, and tune file. The output deck name will be generated automatically, or you may change it to a different name if you wish.

Arbitrary Particle Distributions

If you wish to use an arbitrary particle distribution file as input to your simulated beamline, as opposed to a distribution generated using Twiss parameters, check the “Use Input Particle Distribution File,” box and select the appropriate file. DynacGUI will look by default in the “Particle Distributions” subdirectory of the current folder. (and may break if your file is located at the end of a long path NOT below the current folder)  
  
Your Tune Settings file must still specify a reference A, Q, Energy and starting RF frequency for the tune.

The input file should be formatted as specified in the Dynac manual:

Line 1: [Number of Particles] [unused] [frequency(MHz)]

Lines 2-end: x[cm] x’[rad] y[cm] y’[rad] z[radians] E[MeV] ChargeState

The values in the first line will be ignored in favor of the values in the Tune Settings file. **N.B.:** While Dynac stores z positions as phases relative to the RF frequency, it does NOT read the RF frequency from the distribution file. It is therefore essential that your starting RF frequency matches the frequency at which your distribution file was saved. Failing this will give undesired results. If you need to run a distribution saved at one frequency through a beamline at a different frequency, use the frequency of your *distribution file* as your RF frequency for your tune, and then issue a NEWF command.

1. **Generate deck**

Press the “Generate Deck” button to generate the Dynac input deck file. This file will overwrite the previous file (if any) with the same name. The “View” button will allow you to inspect the generated deck.

1. **Run deck**

Press the “Run Deck” button to run the simulation. In order to run the deck, “dynacv6\_0” must be installed on your system and accessible through the path, or in the local directory. By default, the frontend assumes you have used the MinGW compiler to compile this executable. If you haven’t, uncheck the “MinGW” option box in the lower right hand corner. The output of the Dynac executable is sent to the “Dynac Output” text box, which can be cleared with the “clear” button.

Note that Matlab is unable to recover if Dynac fails to terminate, which can happen in certain beam conditions. If this is the case you will need to manually stop the Dynac process.

1. **Gen/Run for t>tRFQ**

Natively, Dynac will only transport a particle distribution which extends less than one RF period through an RFQ. If particles outside the window are input to an RFQ simulation, the user has the option to either discard the particles or move them, modulo 360 degrees, inside the RF window. When particles are run via the Generate Deck/Run Deck buttons, DynacGUI selects the latter option.

If the user needs to transport a longer distribution, for example to analyze the tails of a buncher longer than a single RFQ period, the Gen/Run for t>tRFQ button will perform the simulation. Since this requires generating a number of separate decks and running Dynac multiple times, not all of the output files will be available. In particular, there will not be a master output deck for the entire line.

What will work: The GUI plot commands for individual plots and the master “envelope/energy plot” button. The Dynac.print and emit.plot files will be correct.

What will NOT work: Dynac.long and Dynac.short files will not be generated, although this capability could in principle be added later. Envelope plots will only display starting from the RFQ.

1. **Plot graphs**

There are two plotting options. To generate a summary plot of the X/Y beam envelope, the particle count, and the reference particle energy vs. position along the beamline, press the “Energy/Envelope Plots” button. This window will give you the option to select or deselect the various data series, and to set the beginning and end point of the plot.

**ReA3 only:** Beam box positions for ReA3 can also be plotted, although selecting the correct beamline for the final positions requires editing the code at present. An offset can be specified for simulations that begin after L016. This option must be activated in the dynacgui.ini file.

The second option allows you to view the individual graphs specified in the layout file. To view individual graphs, click the desired graph in the “Generated Graphs” box. To plot ALL of the generated graphs, click the “plot all” button.

While viewing an emittance plot, click on the plot to autoscale the axes.

The “Tools” submenu of the emittance plots contains several options:

* Show Beam Data – displays basic data, such as TWISS parameters and energy, for the beam at this point. Note that parameters are generated by Dynac from fitting an oval to the particle distribution, so irregular distributions will yield less than helpful results.
* Auxiliary Plots – Allows for the creation of less common plots – X and Y vs Phase and Energy. Also includes an option to plot a histogram of the particles in phase space with an adjustable bin width. A dispersion plot (dp/p vs. x) is also available, and will attempt to fit a dispersion line to the particle data at that point.
* Write Particle Distribution to File (Ctrl+D) – writes a Dynac formatted particle distribution file for the plotted particles to a user selected location. (Default is to the “Particle Distributions” subdirectory of the current directory) This file can later be read using the “Use Input Particle Distribution File” option.
* Export COSY distribution file – exports the current particle distribution file for input into the COSY particle tracking routines written by Mauricio Portillo. Format is:

X(m) / X’(rad) / Y(m) / Y’(rad) / DT (see COSY documentation) / DK (relative energy difference to RP) / DG (relative mass to RP) / DZ (relative charge to RP)

Note that this format does NOT actually store the properties of the reference particle itself.

* Export TRACK distribution file – exports the current particle distribution file for input into TRACK. See the Track documentation for details.

1. **Adjust settings**

It is possible to alter the tune settings directly from the GUI by pressing the “Settings” button. These changes will be reflected in the generated deck, but NOT in the original tune settings file. If the settings have been altered, the settings file name will be marked with an asterisk.

When the window is generated, the units will be populated next to the settings name.

1. **Save A Tune File**

This option allows you to, after editing a tune in the frontend, save a new tune file that can be reloaded later. It will allow you to overwrite old tunes (with a warning) so be careful!

1. **Load a Machine Tune (ReA3 Only)**

To load a machine tune into Dynac, press the “Load ReA3 Machine Tune” button. You will need to specify two ROCS output sets – one for the Q/A line to the end of the L-Line, and one for the L-Line to ANASEN. (At present, the JENSA and AT-TPC lines are not implemented.)

Once the files are loaded, press the “Load Tunes” button. A filename will be automatically generated, and the settings can now be used in Dynac. Note: real settings may produce very different results in simulation than in the actual machine!

This option is disabled by default, and must be enabled by adding the line “ReA3Import<tab>1” to the DynacGUI.ini file.

# Tools

The “Tools” menu contains several potentially useful utilities. Other features may be added by request. The menu itself will only appear if the required scripts are present in the path.

1. **Generate COSY Deck** (gencosydeck.m)

This tool does its best to produce a valid COSY deck for the given tune and layout. Be sure to edit the start of “gencosydeck.m” to conform with your parameters. A few specific limitations:

* + 1. Since DYNAC does not specify apertures for a number of elements, COSY will use a default aperture for these elements.
    2. COSY does not support a number of elements present in DYNAC – these will not be included, but a comment will be added to show where they were left out.
    3. COSY is not a particle tracking code – an envelope to track through the beamline will be generated based on the initial Twiss parameters.

Any deck generated with this tool should probably be treated as a starting point for further refinement.

1. **Generate Scaled Tune** (rescaletune.m)

This will scale the settings in memory for a different Q/A or energy setting. These settings can be used to generate an output deck, but the underling tune file will NOT be saved unless you do so manually. It will ONLY scale beamline elements, but not alter the initial conditions, so be sure to change the input beam accordingly. Also, no allowance is made for change of energy along the line – the entire line is scaled for the same energy change. As such, it may be best to adjust a given line in sections, depending on the individual circumstances.

Note that bending elements whose fields are computed automatically will not be scaled by this routine – they will remain set to “automatic.” Only bending elements which have named tuning parameters present in both the layout and tune files set to manual field values will be scaled.

1. **Save Results**

This option will copy the result files for the most recent simulation run to a separate directory. By default, the created directory will be ./Results/[Deckname], although any directory can be selected. A binary file called “data.mat” will also be created in the directory to store data needed to view the results later.

1. **View Results** (DG\_viewresults.m)

This option opens a utility for viewing previously saved results. Select the directory containing the result files, and then either plot an energy/envelope graph or emittance plots as described above.

1. **Fitting Tool** (DynacGUIFit.m)

This option is detailed below.

1. **Load CS Tune** (loadcs.m)

This option is detailed below.

# View Files Menu

The “View Files” menu allows for quick viewing and editing of the currently selected input files, and selected output files. It attempts to use the default text editor for a given operating system.

It is important to note that this menu will always attempt to load the file, whether or not it was actually generated by the most recent run. If Dynac exits due to lack of remaining particles, only dynac.long will be generated, so be sure you are not looking at the output from a previous run.

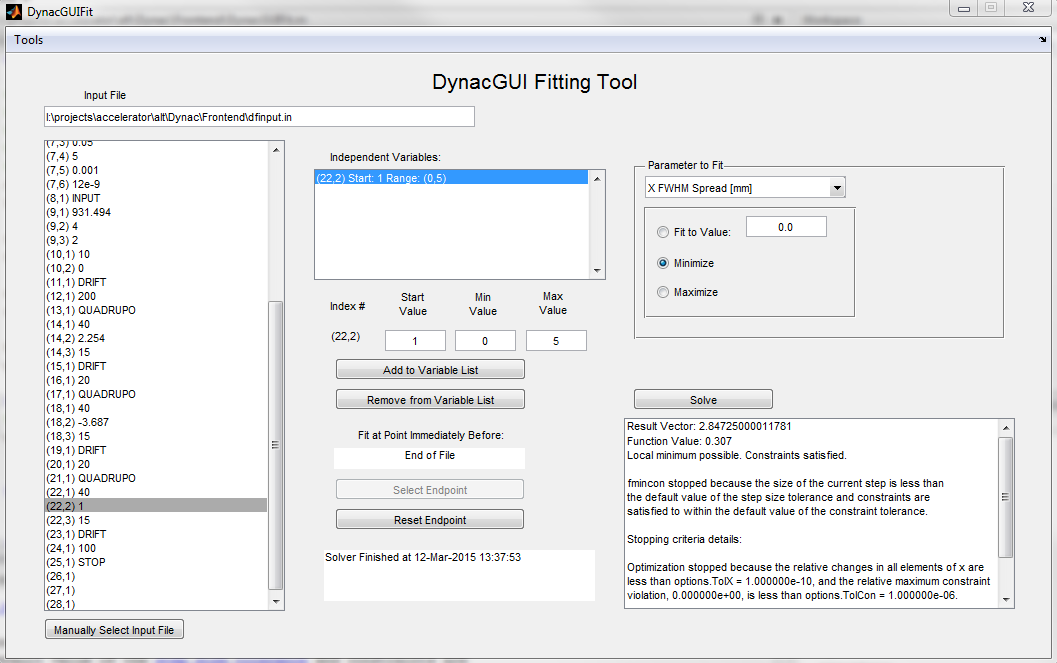
# DynacGUIFit – Fitting Tool

1. **Overview**

Dynac does not in itself include any fitting capability. This tool uses the Matlab Optimization Toolbox in order to tune beamline simulations for certain output conditions. As such, in order to use this tool, you must have a valid license for this toolbox, as well as the DynacGUIFit.m and DynacGUIFit.fig files.

Note: DynacGUI performs a check on launch for an available license and does not display the option to use the fitting tool if the license is not available. Due to the way Matlab handles licenses, it may be necessary to completely quit and relaunch Matlab if a license is checked in after you have started Matlab.

This tool may also be launched independently of DynacGUI. In that case, a fully assembled Dynac deck for the beamline to be fit must be used as an input.



1. **Using DynacGUIFit**
   1. Launch the utility

From the “Tools” menu in DynacGUI, select “Fitting Tool”. The Fitting Tool screen will launch and automatically populate with the currently generated deck. The option to launch the fitting tool will not be available if no deck has yet been generated.

If you wish to use this tool for fitting with an existing Dynac deck, launch ‘DynacGUIFit.m’ directly from within Matlab. (Do not launch ‘DynacGUIFit.fig,’ which will bring up the panel, but with no attendant code.)

* 1. Select the independent variables

After the file is loaded, a list of every numeric variable in the deck will appear in the panel on the left. To choose a variable to be varied for fitting, select it from the list, enter the starting value and the upper and lower limits within which it will be varied, and select, “Add to Variable List.”

If you wish to remove a variable from the list, select it and click “remove variable.” If you wish to change the starting value or limits for a variable already on the list, you may click, “Add to Variable List” a second time, and it will replace the existing entry.

You must select at least one independent variable.

* 1. Select the endpoint (optional)

If you wish to fit for a condition at a point in your input file other than the end, select the element immediately following the point at which you wish to fit, and click the “select endpoint” button. The “reset endpoint” button will return the fitting point to the end of the file.

* 1. Select the fit condition

In the “Parameter to fit” box, the drop down menu will allow you to select which output parameter’s value you wish to optimize. By default, all the values in a Dynac “EMIT” card are available.

Once you have selected the output quantity of interest, you may request the solver either minimize or maximize the value, or approach an arbitrary value as closely as possible. Note that “minimize” means the lowest possible value, including negative numbers. If you wish to minimize the absolute value of a parameter that can take negative values, select “Fit to value,” with a value of zero.

* 1. Run the solver

Once you have selected all of your parameters, click “Solve”. The results will be displayed in the box below the “Solve” button. For now, the results must be manually reentered in DynacGUI, but a future update may provide the capability to return an updated deck to the main program.

* 1. Arbitrary fitting conditions.

If you wish to fit for an arbitrary combination of output parameters, such as alphax+alphy, or number of particles – energy spread, it is possible to manually create your desired fitting function by editing your local copy of DynacGUIFit.m.  
  
At the bottom of the file, create your desired function using the provided “out.radius” function as a guide. Then, near the top of the file, add the name and a description of your function to the ‘matchlist’ and ‘matchnames’ arrays. It is important to add the name and the description at the same point in each list. The locations in the file are clearly marked with comments, and there are more detailed instructions contained in the comments.

# Load CS Tune

The purpose of the loadcs.m module is to allow for the quick importation and calibration of settings from external files, for example control system output files. In order to use this functionality, the loadcs.m file must be visible on the current Matlab path. In addition, an optional calibration file can be used to specify a linear relationship between imported settings and settings used for Dynac. (For example, imported magnet settings in current must be scaled to field via a calibration curve.)

At present, the variable names to be imported must match the names of the parameters in the DynacGUI layout file. This may be updated in future releases if requested.

**Using Load CS Tune**

1. First, set up a calibration file with the values to be imported. The file should consist of three columns. The first column is the name of the variables that need to be calibrated on import. (variables which can be imported unchanged need not be listed) The values in the calibration file will be adjusted using the second and third columns as follows:

Dynac value = column 2 \* imported value + column 3

If a fixed value for a variable is desired regardless of the input parameter, simply set column 2 to zero.

1. Second, add the following lines to “DynacGUI.ini”. If they are not present, system defaults will be used, which can cause unpredictable results for CSsetcol.

Calfile<tab>Default calibration file location

CSsetdir<tab>Default settings file directory (do not use a separator at the end)

CSsetcol<tab>Which column in the settings file contains the data to be imported

1. Once the calibration file is set up and the .ini file modified, run DynacGUI and select “Load CS Tune” from the Tools menu. Use the buttons to select the settings file to be imported and calibration file (if needed), then click the “Load Settings” button to import the settings from the input file. Use the “Edit Tune Settings” button in the main DynacGUI window to verify that the settings have been imported correctly.

Note that any lines in the settings file starting with “#”, “;”, or “-“ will be ignored. If there are header lines without such a delimiter, loadcs will attempt to import them as setting/value pairs. This may or may not cause problems.

# Known Issues:

1. If Dynac should crash for any reason, there will be no notification in the frontend. The process must be identified and killed in the operating system, and the GUI restarted.

# Appendix A - Sample Files

Layout.txt

DRIFT 1.

QUAELEC EQuad2 L030V

DRIFT 1.5

EDFLEC Ebend1

BMAGNET Bbend1

BUNCHER MHB 1 MHB1V MHB1P

BUNCHER MHB 2 MHB2V MHB2P

BUNCHER MHB 3 MHB3V MHB3P

SOLENO Solenoid1 L060V

EMITGR Egraph1 Box 1 Faraday Cup

RFQPTQ RFQ RFQA RFQP

CAVNUM Cav041 CAV1A CAV1P

CAVNUM Cav041 CAV2A CAV2P

CAVNUM Cav085 CAV3A CAV3P

QUADRUPO Quad1 L090F

ENVEL Envel1 Whole Line Envelope

REJECT Aperture1

Devices.txt

EQuad1 15. 5.

EQuad2 10. 5.

Quad1 15 3.75

Bbend1 90 68.1 0 0 0 0 0 0

Ebend1 68.1 45 100000

MHB 2.5

Solenoid1 22.18

RFQ dp\_rea3\_rfq\_ptq\_r10.dat 94 .012 80.5

Cav041 ReA\_field\_041.txt

Cav085 ReA\_field\_085.txt

Egraph1 5 1 40 180 .01

Envel1 5 0 999

Aperture1 5 4000 100 100 2

Tunesettings.txt

A 4

Q 1

RF 80.5E6

Npart 1000

Alphax 0.

Betax .4

Epsx 10.

Alphay 0.

Betay .4

Epsy 10.

Energy .048

Deltae .0001097

L030V 4.0

MHB1V .0007

MHB1P -90

MHB2V .0002457

MHB2P 90

MHB3V .0000805

MHB3P -90

L060V 5.36

RFQA 80

RFQP 0

CAV1A 13.5

CAV1P -90

CAV2A 83.3

CAV2P -20

CAV3A 45

CAV3P -35

L090F -4.583

DynacGUI.ini

Layout Machine Data\ReA3Layout.txt

Devices Machine Data\ReA3Devices.txt

Tune Tune Settings\ReA3\_4He1\_Machine\_QA\_3\_PB\_MHB\_rot.txt

Executable C:\Program Files (x86)\dynacv6p0r11\_w\bin\dynacv6\_0.exe

Executable2 C:\Program Files (x86)\dynacv6p0r13\_b\bin\dynacv6\_0.exe

Executable3 C:\Program Files (x86)\dynacv6p0r13\_c\bin\dynacv6\_0.exe

Mingw 1

ReAImport 0

ReABoxes 0

Edflec 4

# Appendix B – Space Charge

Space charge in DynacGUI is treated as a device which is inserted in the beamline at the point at which space charge computations should be started. In this way, different space charge models can be used at different points in the layout.

There are a several space charge options available from within Dynac, and the user is encouraged to consult the Dynac user’s guide.

The five possible configurations are:

1. HERSC with default parameters

-1 HERSC with manual parameters

2 SCHERM

3 (0) SCHEFF with default parameters

3 (1) SCHEFF with manual parameters

To invoke each of these, add the following lines to your **device file**. [name] is a name you assign to that set of space charge parameters. “Flag” is a flag which determines which elements are calculated with space charge on. (1 = all but drifts, 2 = accelerating elements only, 3= all elements) For definitions of the other parameters, see the Dynac manual.

HERSC with default parameters

[name] 1 flag RCDF

HERSC with manual parameters

[name] -1 flag LMAXI MMAXI NMAXI FXRMS FYRMS FZRMS

SCHERM

[name] 2 flag

SCHEFF with default parameters

[name] 3 flag 0

SCHEFF with manual parameters

[name] 3 flag 1 SCE(2) SCE(3) SCE(4) SCE(5) SCE(6) SCE(7) SCE(9)

To your **layout file** add:

SCDYNAC [name] [Current\_name]

To your **tune settings file** add:

[Current\_name] [beam current (mA)]

You may set up as many space charge configurations as you wish with different [name]s, and you may invoke them at any point in the beamline with the SCDYNAC command. [Current\_name] is the name of a tunable parameter. If you wish to set different beam currents at different points in your layout, use SCDYNAC at each location where the current should change, using a different variable name for [current\_name] at each location.

Cavity Computation Location:

To change the position within cavities where space charge is computed, add the following line to your **layout file:**

SCPOS [fraction]

[fraction] is a number between zero and one which defines the position relative to the length of the cavity where space charge is computed. The default value which is used if no fraction is specified is 0.5, the center of the cavity.

Space Charge Within Bending Magnets:

To compute space charge in a bending magnet, Dynac recommends dividing the magnet into two separate entries in the layout, and placing the computation between them. To invoke space charge calculation for a bending magnet with central trajectory of length L, use the SCDYNEL command as follows:

BMAGNET ….. (parameters for a bend magnet of length L/2)

SCDYNEL L [cm]

BMAGNET …. (parameters for a second bend magnet of length L/2)

Be sure to also invoke the SCDYNAC command earlier in your layout. For more detail on how to use space charge through a bending magnet, please see the Dynac manual.

Example Files:

**Device File:**

Bbend1 90 68.1 0 0 0 0 0 0

Bbend2 45 53.49 0 0 0 0 0 0

Cav010 Machine Data/HWR010.txt

Solenoid1 6.0

scharge1 3 2 1 2 3 4 5 6 1

Comment:

Only one type of space charge calculation is defined here, type 3 (SCHEFF) with manual setting of parameters.

**Layout File:**

SCDYNAC scharge1 CURRENT1

DRIFT 1.5

BMAGNET Bbend1

DRIFT 23.0

SOLENO Solenoid1 BS1

DRIFT 14.5

CAVNUM Cav010 CAV1\_AMPL CAV1\_PHASE

DRIFT 23.0

SOLENO Solenoid1 BS2

DRIFT 14.5

SCPOS 0.25

CAVNUM Cav010 CAV2\_AMPL CAV2\_PHASE

DRIFT 14.5

SCDYNAC scharge1 CURRENT2

BMAGNET Bbend2

SCDYNEL 106.97

BMAGNET Bbend2

DRIFT 14.5

Comments:

Space charge is invoked from the start of the line with beam current value set by CURRENT1 in the tune file. At the second SCDYNAC card, the beam current changes to CURRENT2, also set in the tune file. Since the same space charge definition (scharge1) is used, all other parameters are the same.

Space charge is calculated at the center of the first cavity, and then moved to the 0.25\*Length for the second cavity by the SCPOS card.

This line contains two bending magnets of angle 90 degrees and total path length 106.97. For the first magnet, no space charge is calculated, and the magnet is defined as one element. For the second one, the magnet is defined as two magnets of half the length, and space charge is invoked at the center with the SCDYNEL card, applied over the length of both magnets.

**Tune Settings File:**

A 1

Q 1

RF 162.5E6

Npart 1000

Alphax -0.58585

Betax .58073

Epsx 13.68

Alphay -0.58585

Betay .58073

Epsy 13.428

Energy 2.12306

ZLaw 1

Alphaz -.125

Betaz 2.3

Epsz 10.52

CURRENT1 1.5

BS1 32.2444

CAV1\_AMPL 56.438

CAV1\_PHASE 60.0

BS2 33.3262

CAV2\_AMPL 61.786

CAV2\_PHASE 80.0

CURRENT2 1

References:

*PROGRAM DYNAC User’s Guide*, P. Lapostolle, S. Valero, E. Tanke, CERN, 6-Aug-2014

*COSY INFINITY Version 9.0 Beam Physics Manual,* M. Berz and K. Makino, MSU Report MSUHEP 060804, August 2006

*TRACK – a Code for Beam Dynamics Simulation in Accelerators and Transport Lines with 3D Electric and Magnetic Fields,* P.N. Ostruomov et al., ANL Physics Division, March 29, 2007