JSPEC v. 1.0.0 User Manual

Using the text-based user interface

Run JSPEC with the input file

Put the input file in the same directory with the JSPEC program and run JSPEC as :

> jspec.exe *inputfile*

Format of the input file

The input file is a plain text file and it will be parsed by the program line by line. Each command or expression should occupy a separate line. Comments start with "#". Everything behind the "#" in the line will be ignored by the program. Blank lines, white spaces and tabs are also ignored. The input file is NOT case-sensitive.

The input file is organized by various sections. All the sections fall into three different categories: (1) scratch section, (2) definition sections and (3) operation section. All the sections with the respective categories and usages are listed in the following table.

Section name	Category	Usage
section_scratch	scratch	Define variables and do calculations with the variables. The variables defined in this section can be used in definition sections.
section_ion	definition	Set parameters for the ion beam
section_ring	definition	Set parameters for the ion ring
section_e_beam	definition	Set parameters for the cooling electron beam
section_cooler	definition	Set parameters for the cooler
section_ibs	definition	Set parameters for IBS rate calculation
section_ecool	definition	Set parameters for electron cooling rate calculation
section_run	operation	Create the objectives (ion beam, ion ring, electron beam, cooler) and perform the calculation and/or the simulation.

The input file starts with a section by calling the section name. Once a section name is called, the respective section is created, and this section ends when another section name is called or when the input file ends. Sections can be repeated called and the latter one overwrite the previous ones. But if a parameters is not set again in the latter one, its value remains.

The following example includes three different sections in three different categories.

```
section_scratch #scratch section
        m = 938.272
        ke = 8000
        gamma = ke/m + 1
        print gamma
        list_const
section_e_beam #definition section, define the parameters for electron beam
        gamma = gamma
        tmp_tr = 0.1
        tmp_l = 0.1
        shape = dc_uniform
        radius = 0.004
        current = 2
section_run #operation function
        create_e_beam
```

The first section is a scratch section. In this section, three variables, m, ke, and gamma, are defined. The values of m and ke are assigned, and gamma is calculated from ke and m. The calculation is supported by the math parser, muParser. Fundamental calculations and functions are supported, including summation, subtraction, multiplication,division, square root, exponential function, etc. For more details about the muParser, please refer to http://beltoforion.de/article.php?a=muparser . The command "print gamma" will print the value of gamma to the screen. The following command will show a list of all the constant variables supported by the scratch section on the screen. All the constant variables and their values are listed in the following table.

Constant	Value	Meaning
k_c	299792458.0	speed of light, in m/s
k_e	1.602176565E-19;	Elementary charge, in C
k_pi	3.1415926535897932384626	π
k_u	931.49406121	Atomic mass unit, in MeV/c^2
k_me	0.510998928	Electron mass, in MeV/c^2
k_ke	8.9875517873681764E+9	Coulomb's constant, in N*m^2/C^2

The second section is a definition section, which sets the parameters for the cooling electron beam. In all the expressions in this section, the left side of the "=" sign is a keyword in section_e_beam, which corresponds to a parameter of the electron beam, and the right side is the valued assign to the keyword (the parameter). The first expression in this section is "gamma = gamma".[1] The left gamma is a keyword, which represents the Lorentz factor of the electron beam. The right gamma is the variable defined in the above scratch section. This expression assigns the value of the scratch variable gamma to the keyword gamma. Please note that a scratch variable can be used in other sections to set the value for a keyword, but a keyword cannot be used in the same way. A keyword should always be on the left side of the "=" sign. This is the most important difference between a scratch variable and a keyword. The following expressions assign values for other parameters of the electron beam, which are the transverse temperature, the longitudinal temperature, the shape, the radius and the current of the electron beam respectively. Depending on the shape of the electron, various parameters need to be set. In this example, one needs to set the radius and the current for a uniform DC electron beam. Other supported shapes and the related parameters (keywords) can be found in the lists in the next chapter.

The third section is the operation section. In the operation section, one can create the objects of the elements, calculate the expansion rate and perform the simulation. In this example, we create an object of the electron beam that has been defined in the above definition section. Please note that the definition section only records the values of the parameters, an element will not be created until the respective command is called in the operation section. For more commands supported in the operation section, please check out the list in the next chapter.

IBS Expansion Rate Calculation

To calculate the IBS expansion rate, one needs to define the ion beam and the ring. Then set the parameters for IBS rate calculation. Finally, in the operation section create the ion beam and the ring, and call the command to calculate the IBS expansion rate.

```
section_ion             # Define the ion beam
        ......
section_ring    # Define the ring
        ......
section_ibs             # Set parameters for IBS rate calculation
        ......
section_run
        create_ion_beam         # Create the ion beam
        create_ring                     # Create the ring
        calculate_ibs           # Calculate the IBS rate
```

To calculate the total expansion rate, which is the summation of the IBS expansion rate and the electron cooling rate, one can call the command "total_expansion_rate" in section_run.

Cooling Rate Calculation

To calculate the cooling rate, one needs to define the ion beam, the ring, the electron beam and the cooler. Then set the parameters for cooling rate calculation. Finally, in the operation section create all the related elements aforementioned and call the command to calculate the cooling rate.

```
section_ion             # Define the ion beam
        ......
section_ring    # Define the ring
        ......
section_e_beam  # Define the electron beam
        ......
sectoin_cooler  # Define the cooler
        ......
section_ecool   # Set the parameters for the electron cooling rate calculation
        ......
section_run
        create_ion_beam         # Create the ion beam
        create_ring                     # Create the ring
        create_e_beam           # Create the electron beam
        create_cooler           # Create the cooler
        calculate_ecool         # Calculate the electron cooling rate
```

Simulation

One can simulate the evolution of the ion beam under the IBS effect and/or electron cooling effect during a predetermined time. The emittances, momentum spread, bunch length (for bunched ion beam), and the total expansion rate in all the three dimensions will be outputted into a text file. If desired, the coordinates of all the ion samples can also be saved into files. These parameters are set in section_simulation, and the simulation starts when the command "run_simulation" is called in section_run.

```
section_simulation  # Set the parameters for the simulation
        ......
section_run
        run_simulation  # Start simulation
```

Keep a constant bunch length of the ion beam in simulation

The momentum spread of the ion beam changes due to the intrabeam scattering effect and the electron cooling effect during the simulation, hence the bunch length changes if the RF voltage is constant. However, if the RF voltage changes accordingly with the momentum spread, it is possible to maintain a constant bunch length. JSPEC allows the user to choose whether to keep the bunch length constant in the simulation. When the bunch length is maintained constant, the RF voltage is calculated and saved in the output file.

To use this feature, one needs to set the parameter "fixed_bunch_length" in section_simulation to be **true**. One also needs to set the parameters, rf_h (harmonic number), rf_phi (RF phase), and gammar_tr (transition gamma) in section_ring.

```
section_ring #define the ring
        ...
        rf_h = 3584
        rf_phi = 0
        gamma_tr = 12.46

...

section_simulation
        ...
        fixed_bunch_length = true
```

List of sections, keywords, and commands

section_scratch

Keywords	Meaning
list_var	list all the variables that has been defined.
list_const	list all the constants
list_exp	list all the expression
print	Use this command in format "print x" and it will print the value of the variable x in the screen

section_ion

Keywords	Meaning
charge_number	Number of the charges of the ion
mass	Mass in [MeV/c^2] of the ion
kinetic_energy	Kinetic energy in [MeV] of the ion
norm_emit_x	Normalized horizontal emittance in [m*rad] of the ion beam
norm_emit_y	Normalized vertical emittance in [m*rad] of the ion beam
momentum_spread	Momentum spread of the ion beam
particle_number	Total particle number for coasting ion beam or the particle number of one bunch for bunched ion beam.
rms_bunch_length	RMS bunch length for bunched ion beam in [m]

section_ring

Keywords	Meaning
lattice	The name of the file that saves the lattice. This file should be in the MAD X output format (.tfs).
qx	Transverse betatron tune
qy	Vertical betatron tune
qs	Synchrotron tune
gamma_tr	Transition gamma
rf_v	Voltage of the RF cavity in [V]
rf_h	Harmonic number
rf_phi	RF phase in [2π]

section_cooler

Keywords	Meaning
length	Length of the cooler in [m]
section_number	Number of the coolers
magnetic_field	Magnetic field in [T]
bet_x	Beta function in horizontal direction in [m]
bet_y	Beta function in vertical direction in [m]
disp_x	Dispersion in horizontal direction in [m]
disp_y	Dispersion in vertical direction in [m]
alpha_x	Alpha in horizontal direction
alpha_y	Alpha in in vertical direction
disp_dx	Derivative of the dispersion in horizontal direction
disp_dy	Derivative of the dispersion in vertical direction

section_e_beam

Keywords	Meaning
gamma	Lorentz factor gamma for the cooling electron beam
tmp_tr	Transverse temperature in [eV]
tmp_l	Longitudinal temperature in [eV] for the cooling electron beam
shape	Electron beam shape. Choose from dc_uniform, bunched_gaussian, bunched_uniform, bunched_uniform_elliptic, dc_uniform_hollow, bunched_uniform_hollow, bunched_user_defined.
radius	Radius of dc_uniform or bunched_uniform electron beam in [m].
current	Current of dc_uniform or bunched_uniform electron beam. For bunched_uniform beam, set the current as if it is a dc_uniform beam in [A].
length	Length of the bunched_uniform electron beam in [m].
sigma_x	RMS size in horizontal direction of bunched_gaussian electron beam in [m].
sigma_y	RMS size in vertical direction of bunched_gaussian electron beam in [m].
sigma_z	RMS bunch length of bunched_gaussian electron beam in [m].
rh	Length of the semi-axis in horizontal direction in [m].
rv	Length of the semi-axis in vertical direction in [m].
r_inner	Inner radius of a hollow beam in [m]
r_outter	Outter radius of a hollow beam in [m]
particle_file	Name of the file that saves the particles if the beam shape is defined as "bunched_user_defined"
total_particle_number	Total number of particles to load from the user-provided file
box_particle_number	Maximum number of particles in each childless box when constructing the tree structure. Default is 200.
line_skip	Number of lines to skip when loading particles from the user-provided text file.
vel_pos_corr	Whether to consider the correlation between the velocity and the position. Default is false.
binary_file	Whether the user-provided file is in binary format. Default is false, which means a text file.
buffer_size	Buffer size when loading particles from the user-provided binary file.

section_ibs

Keywords	Meaning
nu	Set the grid number in horizontal direction for the 3D integration.
nv	Set the grid number in vertical direction for the 3D integration.
nz	Set the grid number in longitudinal direction for the 3D integration.
log_c	Coulomb logarithm. If log_c is set, then the integration in the longitudinal direction is replaced by the Coulomb logarithm. Thus the parameter nz is ignored.
coupling	Transverse coupling rate, ranging from 0 to 1.
model	Model for IBS expansion rate calculation: Martini or BM.

section_ecool

Keywords	Meaning
sample_number	Number of the sample ions.
force_formula	Choose the formula for friction force calculation. Now only support the Parkhomchuk formul, using force_formula = PARKHOMCHUK.

section_simulation

Keywords	Meaning
time	Total time to simulate, in [s].
step_number	Total number of steps. The time interval of each step is time/step_number.
sample_number	Number of the sample ions. The parameter must be set when using the Particle model to simulate the IBS expansion process without cooling. When setting this parameter with cooling effect, the "sample_number" parameter in the "section_ecool" will be overwritten by this value.
ibs	Choose to simulate the IBS effect or not by setting the value as "true" or "false".
e_cool	Choose to simulate the electron cooling effect or not by setting the value as "true" or "false".
model	"RMS" or "Particle" model to choose for the simulation.
output_file	Output file name
output_interval	The interval of steps to write into the output file. Default is one.
save_particle_interval	The interval of steps to save the 6D coordinates of the ions. No saving if the value is less than zero. Default is -1. This is only useful when using the Particle model in simulations.
ref_bet_x	TWISS parameters for the reference point. Only needed when the "model beam" method is selected and the electron cooling effect is not included in the simulation.
ref_bet_y	Same as above.
ref_alf_x	Same as above.
ref_alf_y	Same as above.
ref_disp_x	Same as above.
ref_disp_y	Same as above.
ref_disp_dx	Same as above.
ref_disp_dy	Same as above.
fixed_bunch_length	Maintain a constant ion bunch length. Default is false.

section_run

Keywords	Meaning
create_ion_beam	Create the ion beam.
create_ring	Create the ring. Must create the ion beam before calling this command.
create_e_beam	Create the electron beam
create_cooler	Create the cooler.
calculate_ibs	Calculate the IBS rate and output to the screen. Must create the ion beam and the ring before calling this command.
calculate_ecool	Calculate the electron cooling rate and output to the screen. Must create the ion beam, the ring, the electron beam, and the cooler before calling this command.
total_expansion_rate	Calculate the total expansion rate (summation of the ibs rate and electron cooling rate) and output to the screen. Must create the ion beam, the ring, the electron beam, and the cooler before calling this command.
run_simulation	Simulate the evolution of the ion beam under IBS and/or electron cooling effect(s).

Example

In the following example, a DC electron cooler and a bunched proton beam is defined. The IBS rate and the electron cooling rate are calculated. Then the evolution of the proton beam under both the IBS effect and the electron cooling effect is simulated for 600 seconds.

```
section_ion                             # Define the ion (proton) beam
        charge_number = 1       # Charge number
        mass = 938.272          # Mass of the ion
        kinetic_energy = 8000   # Kinetic energy
        norm_emit_x = 2.2e-6    # Normalized emittance in horizontal direction
        norm_emit_y = 2.2e-6    # Normalized emittance in vertial direction
        momentum_spread = 0.0006        # Momentum spread
        particle_number = 6.58e11       # Total ion number (per bunch)
        rms_bunch_length = 7            # Rms bunch length of the bunched ion beam
section_ring                                                            # Define the ring
        lattice = MEICColliderRedesign1IP.tfs   # file that saves the lattice of the ring
section_ibs #define the arguments for IBS calculation
        nu = 100                # Grid number in horizontal direction for IBS integration
        nv = 100                # Grid number in vertial direciton for IBS integration
        nz = 40                 # Grid number in longitudinal direction for IBS integration
        log_c = 20.6    # Define Coulomb logrithm. nz is ignored after log_c is defined.
        coupling = 0    # No coupling
section_cooler                          # Define the cooler
        length = 3.4                    # Cooler length
        section_number = 1              # Number of coolers
        magnetic_field = 0.039  # Magnetic field
        bet_x = 10                              # Twiss parameter at the cooler
        bet_y = 10
        #disp_x = 0                             # If the values are zero, the command can be
omitted.
        #disp_y = 0
        #alpha_x = 0
        #alpha_y = 0
        #disp_dx = 0
        #disp_dy = 0
section_scratch                         # A scratch section
        m = 938.272                             # Define variable m and assign a value.
        ke = 8000                               # Define variable ke and assign a value.
        gamma = ke/m + 1                # Define variable gamma and calculate its value.
section_e_beam                          # Define the electron beam
        gamma = gamma                   # Lorentz factor, the right "gamma" is the variable define
above.
        tmp_tr = 0.1                    # Transverse temperature
        tmp_l = 0.01                    # Longitudinal temperature
        shape = dc_uniform              # Shape of the electron beam, DC beam with uniform charge
density
        radius = 0.004                  # Radius of the DC electron beam
        current = 2                             # Current is 2 A
section_ecool                                           # Set parameters for electron cooling rate
calculation
        sample_number = 10000                   # Number of ion samples
        force_formula = PARKHOMCHUK             # Formula for friction force calculation
section_run                                             # Operation section
        create_ion_beam
        create_ring
        calculate_ibs                           # Calculate the IBS rate
        create_e_beam
        create_cooler
```

```
        calculate_ecool                         # Calculate the electron cooling rate
        total_expansion_rate           # Calculate the total rate = IBS rate + electron cooling
rate
section_simulation                                                  # Set parameters for
simulation
        ibs = on                                                            # Simulate ISB
effect
        e_cool = on                                                         # Simulate
electron cooling effect
        time = 600                                                          # Time to simulate
        step_number = 600                                           # Number of steps
        sample_number = 100000                                  # Number of ion samples
        #save_particle_interval = 100                   # Save the coordinates of the ions every
100 steps
        output_file = simulation_test.txt               # File to save the simulation results
        model = particle                                            # Select the model used in
the simulation
section_run                                             # Operation section
        run_simulation                          # Start simulation
```

1. The author intended to write this expression in this way in order to emphasize the difference between a scratch variable and a keyword. However, this expression may be confusing. So it is not recommended to use scratch variables with the same name of a keyword. ↩