



- Here a short demo on how to run the **PSPA** platform on the LAL server
- To be able to run it please contact us at:

pspa@lal.in2p3.fr

The web interface: login

- Login credentials to be requested

- Server address:

http://pspa-dev.lal.in2p3.fr/?_=/pspa



Login

User name

Enter your user name

Password

Enter your password

Remember me

Keeps login for 2 weeks

Login

[Lost password](#) | [Register](#)

“Main” window


- Here you can:
 - Import or export existing files (from the server or from your computer)
 - Build a lattice from scratch
 - Add/remove elements
 - Modify element’s properties



Accelerator elements

GUN	BEAM	DRIFT
BEND	RF CELL	SOLEN
FIT	MARK	MULT
QUAD	SEXT	TR WAVE

sector machine: pspa run

sector1 

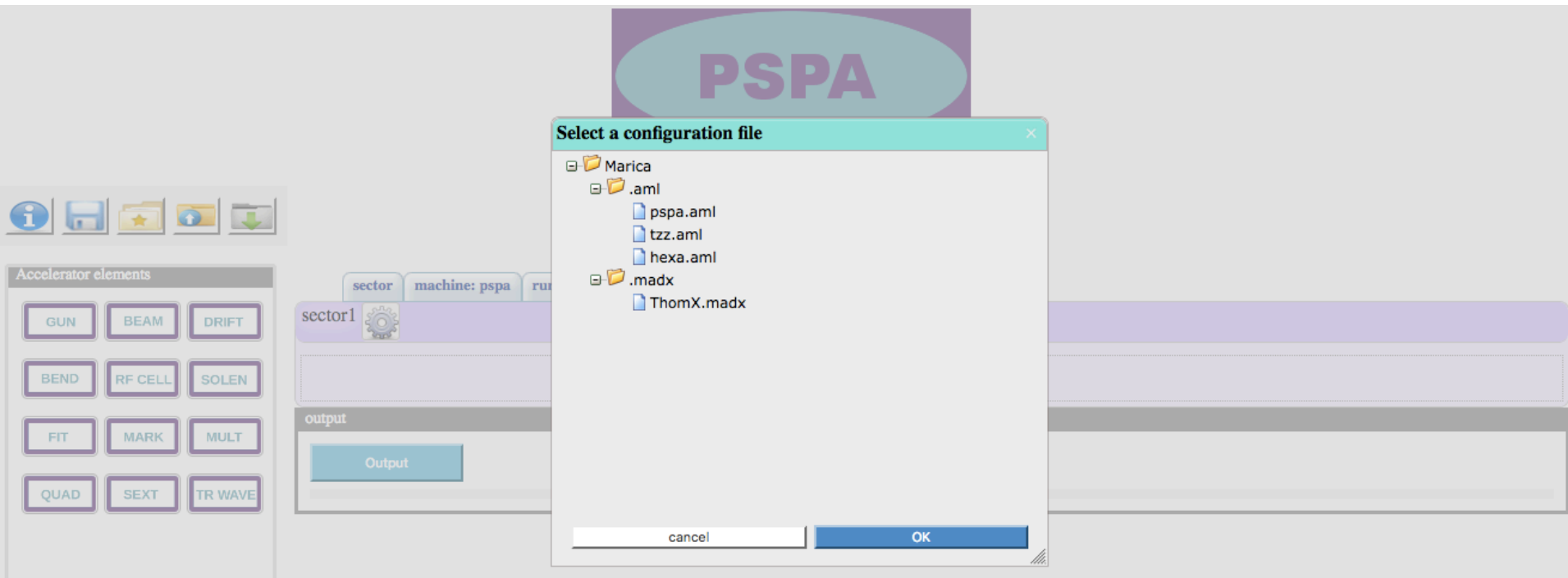
Drag elements here....

output

Output

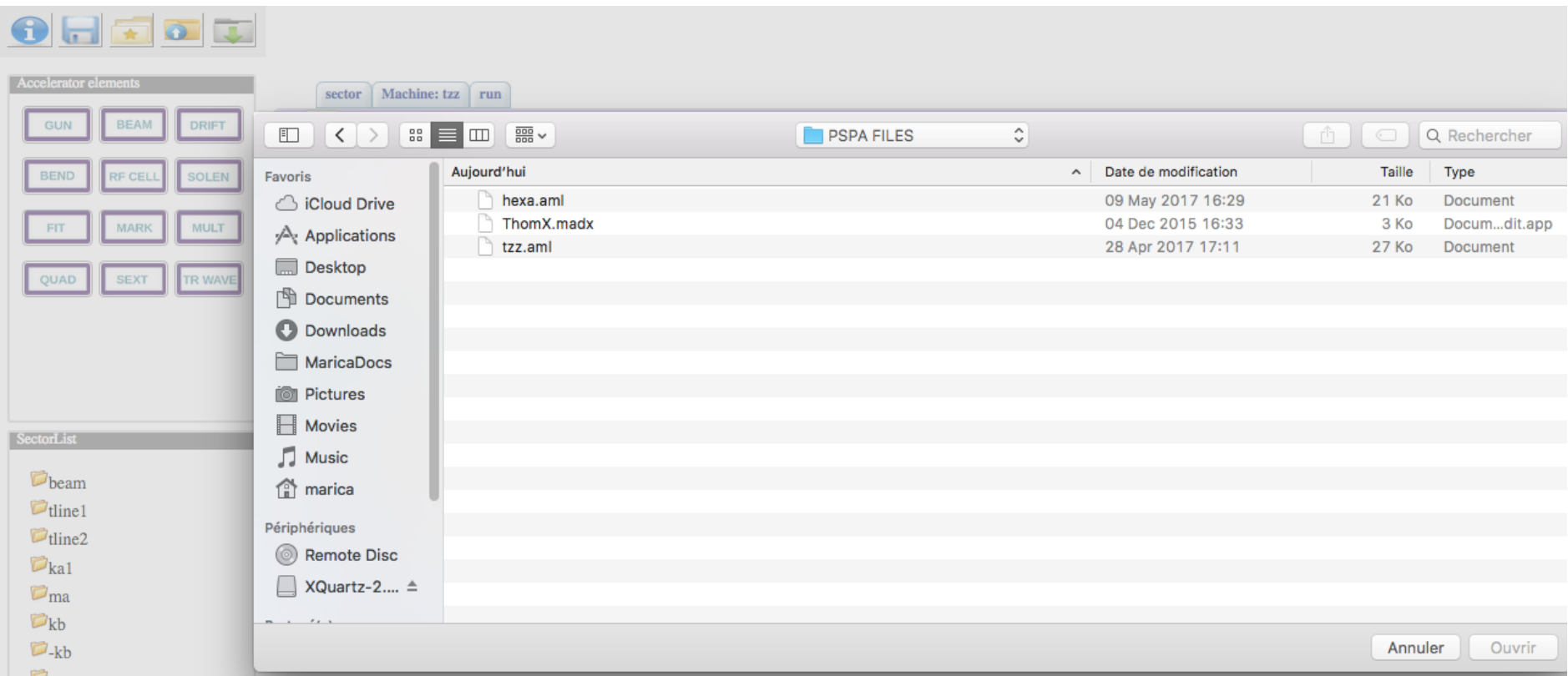
Select input file

- Files format can be: .aml, madx,
- In the example below files are on the user's area on server



Select input file

- In the example below files are on the user's computer



Saving input/output files

- In the example below files can be saved from the server to the user's computer

The screenshot displays the PSPA (Particle Simulation and Parameter Analysis) software interface. A central dialog box titled "Select the file to download" is open, showing a file tree structure under the user "Marica". The files listed are:

- pspa.aml
- ThomX.madx
- ThomX-input.1.1.madx
- thomx-image.1.ps
- madx.ps
- ThomX-output.1.1.madx
- pspa-input.2.1.madx
- pspa-twiss.2.1.madx
- pspa-image.2.ps

The background interface includes a "SectorList" on the left with folders for "beam", "tline1", and "tline2". The main area shows "my accelerator" configuration with "Machine: tzz" and "run" buttons. Below this, "computing blocks" are defined with "FROM" and "TO" fields. The "output" section shows a log of a successful run: "run /usr/local/PSPA/PSPA/pspaWT/working/beta finished normally". On the right, there are "SOFTWARE:" dropdown menus for "beta" and "madx", each with a "run number" field set to "2".

Managing files and element's properties

1. Download file from Server
2. Upload file to Server
3. Import file from user's space
4. Export file from user's space



Lattice elements

Accelerator elements

QUN	BEAM	DRIFT
BEND	RF CELL	SOLEN
FIT	MARK	MULT
QUAD	SEXT	TR WAVE

sector Machine: hexa run

ka2

DRIFT	BEND	DRIFT	QUAD	DRIFT	QUAD	DRIFT	BEND
dr6	kick2	dr6	qp2	dr6	qp1	dr8	sept2

output

restore Lattice:: hexa don

element bend

label of the element : kick2

angle (rad) : -0.011

length (m) : 0

radius (m) : -16.1602

momentum (MeV/c) : 0


aperture (cm) : 1e-06

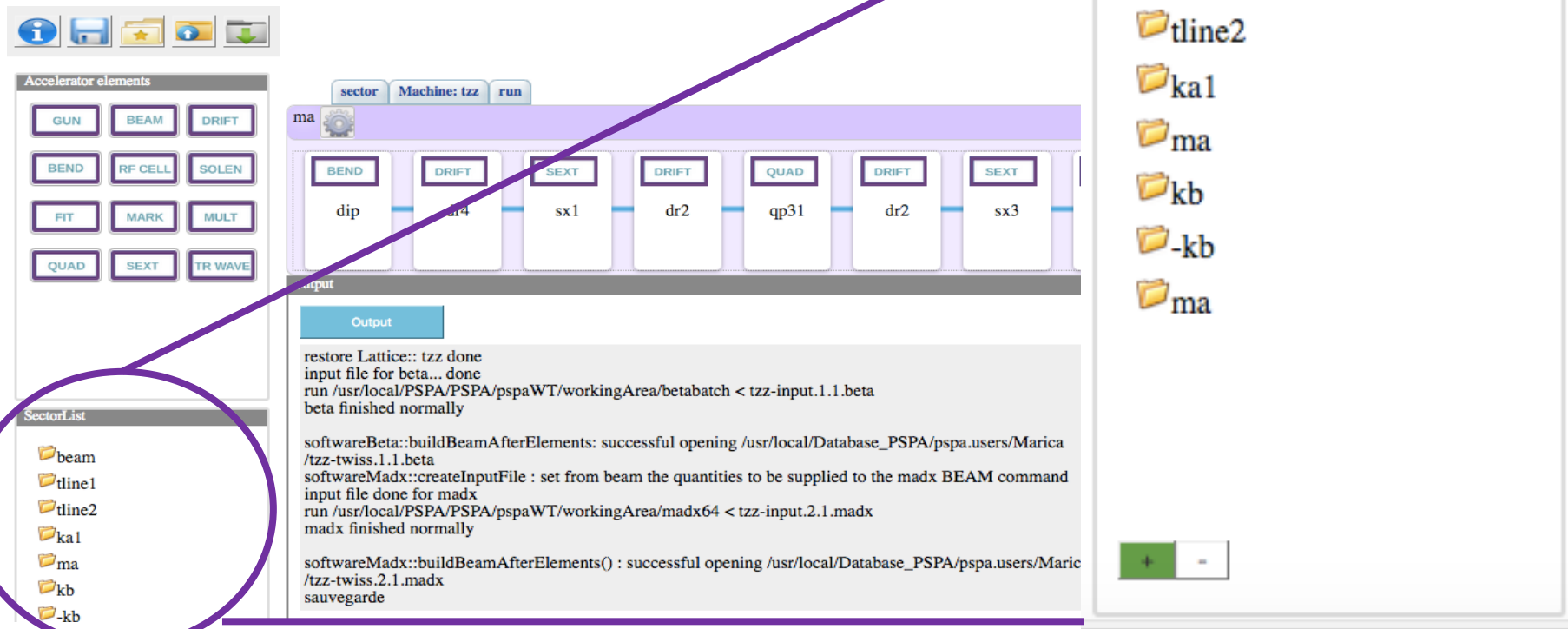
pole face rotations (rad) entrance : 0 exit : -0.011

fringe field integral : 0 half gap (m) : 0

OK Cancel

Select lattice or part of it

- In the example below a lattice has been downloaded and element's properties can be managed
- The "Sector list" on the bottom left contains the lattice sections as defined by MADX "line" command
- It is possible to add/remove them easily by hitting 



Accelerator elements

sector Machine: tzz run

ma

BEND DRIFT SEXT DRIFT QUAD DRIFT SEXT

dip dr4 sx1 dr2 qp31 dr2 sx3

Output

```
restore Lattice:: tzz done
input file for beta... done
run /usr/local/PSPA/PSPA/pspaWT/workingArea/betabatch < tzz-input.1.1.beta
beta finished normally

softwareBeta::buildBeamAfterElements: successful opening /usr/local/Database_PSPA/pspa.users/Marica
/tzz-twiss.1.1.beta
softwareMadx::createInputFile : set from beam the quantities to be supplied to the madx BEAM command
input file done for madx
run /usr/local/PSPA/PSPA/pspaWT/workingArea/madx64 < tzz-input.2.1.madx
madx finished normally

softwareMadx::buildBeamAfterElements() : successful opening /usr/local/Database_PSPA/pspa.users/Maric
/tzz-twiss.2.1.madx
sauvegarde
```

SectorList

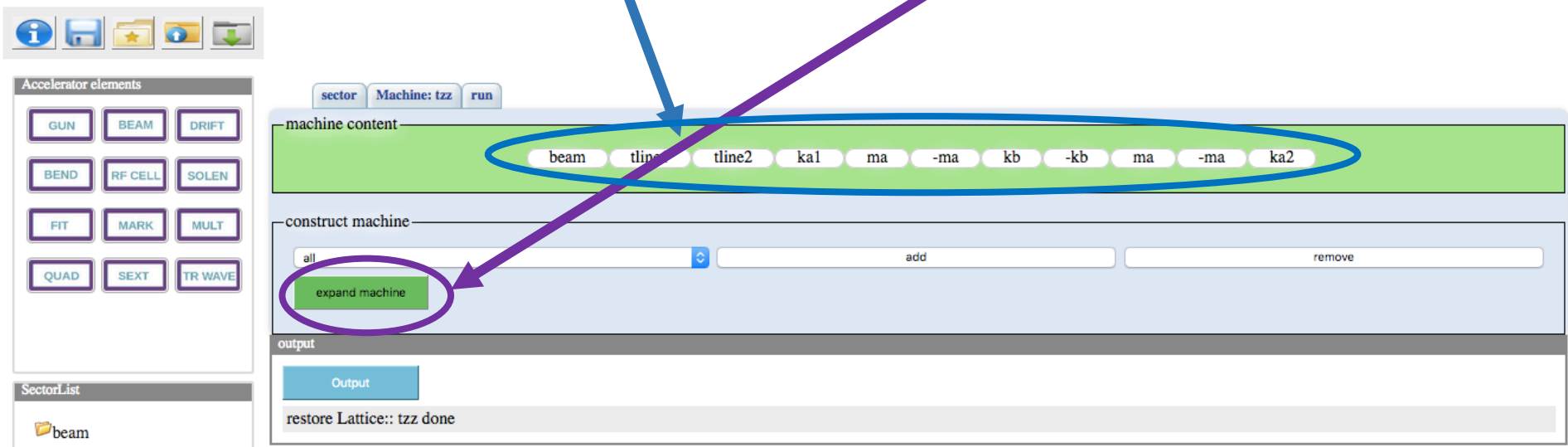
- beam
- tline1
- tline2
- ka1
- ma
- kb
- kb
- ma

SectorList

- beam
- tline1
- tline2
- ka1
- ma
- kb
- kb
- ma

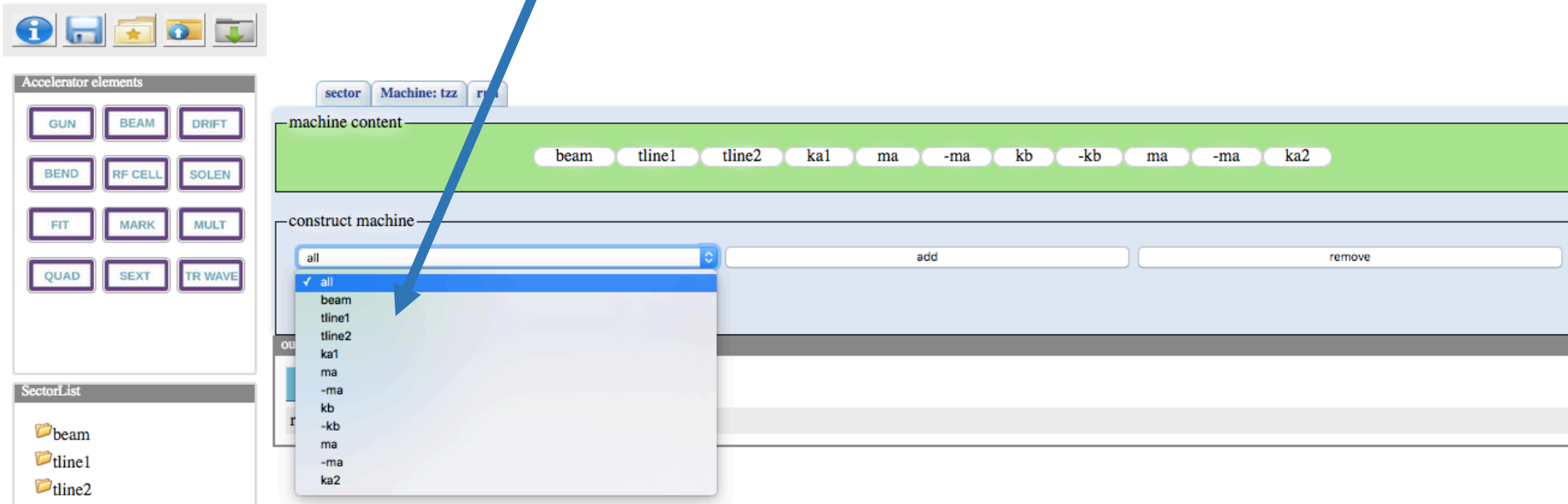
“Build lattice” window

- Once the lattice has been created, in this window you can manage it for execution, by adding/deleting part of it or select “sectors” to be simulated
- Once selected your lattice hit the “expand machine” button



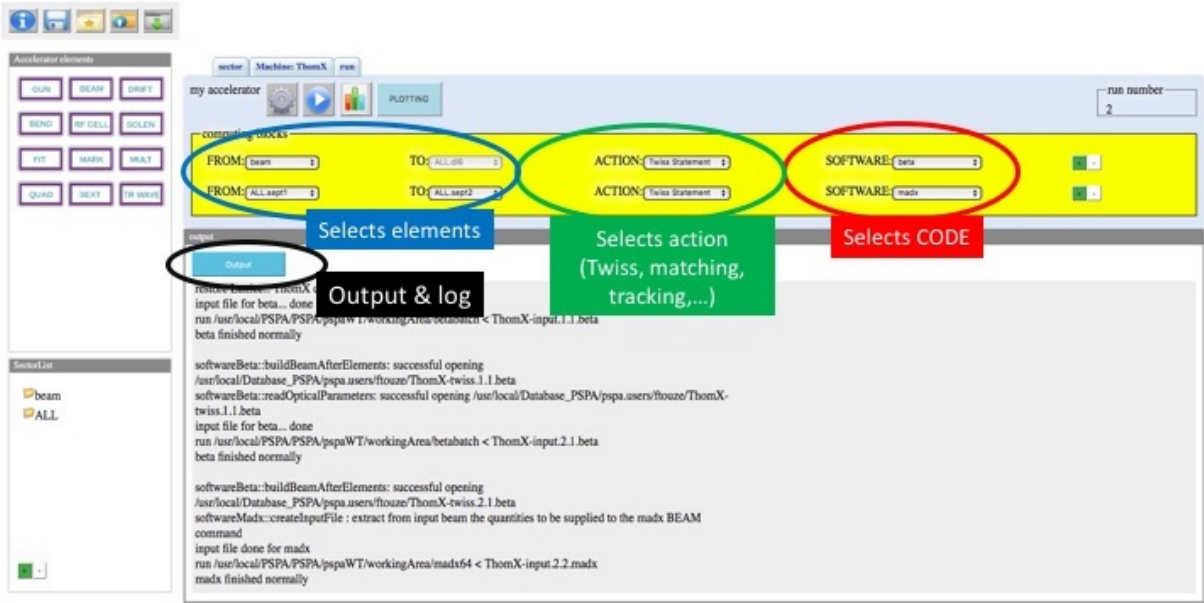
“Build lattice” window

- Select a specific sector of your lattice



“RUN” window

- In this window you can select “actions” and “codes”
- Hitting the **+** button it is possible to select different sectors to be simulated with different codes (example below shows calculation of Twiss functions for **ThomX Transfer Line** with **BETA** and **ThomX Ring** with **MADX**)



“RUN” window

- Select “sector”



The screenshot shows the PSPA software interface. On the left is a sidebar with 'Accelerator elements' (GUN, BEAM, DRIFT, BEND, RF CELL, SOLEN, FIT, MARK, MULT, QUAD, SEXT, TR WAVE) and a 'SectorList' section. The main window has tabs for 'sector', 'Machine: tzz', and 'run'. Below the tabs are icons for settings, play, and plotting, and a 'PLOTING' button. A 'run number' field shows '0'. The 'computing blocks' section is highlighted in yellow and contains a table with columns for FROM, TO, ACTION, and SOFTWARE. The 'FROM' field is set to 'beam.b00' and the 'TO' field is set to 'sept2', both of which are circled in purple. The 'ACTION' field is set to '-- none --' and the 'SOFTWARE' field is set to 'unknownSoftware'. Below this is an 'output' section with an 'Output' button and a log entry: 'restore Lattice:: tzz done'.

FROM	TO	ACTION	SOFTWARE
beam.b00	sept2	-- none --	unknownSoftware

“RUN” window

- Select “action”




Accelerator elements

GUN	BEAM	DRIFT
BEND	RF CELL	SOLEN
FT	MARK	MULT
QUAD	SEXT	TR WAVE

Sector List

- beam

sector Machine: tzz run

my accelerator    PLOTTING run number 0

computing blocks

FROM: beam.b00 TO: sept2 ACTION: -- none -- SOFTWARE: unknownSoftware

output

restore Lattice:: tzz done

Note: The ACTION dropdown menu is open, showing options: Twiss Statement, Matching Module, Particle Tracking, Error Definitions, and -- none -- (selected).

“RUN” window

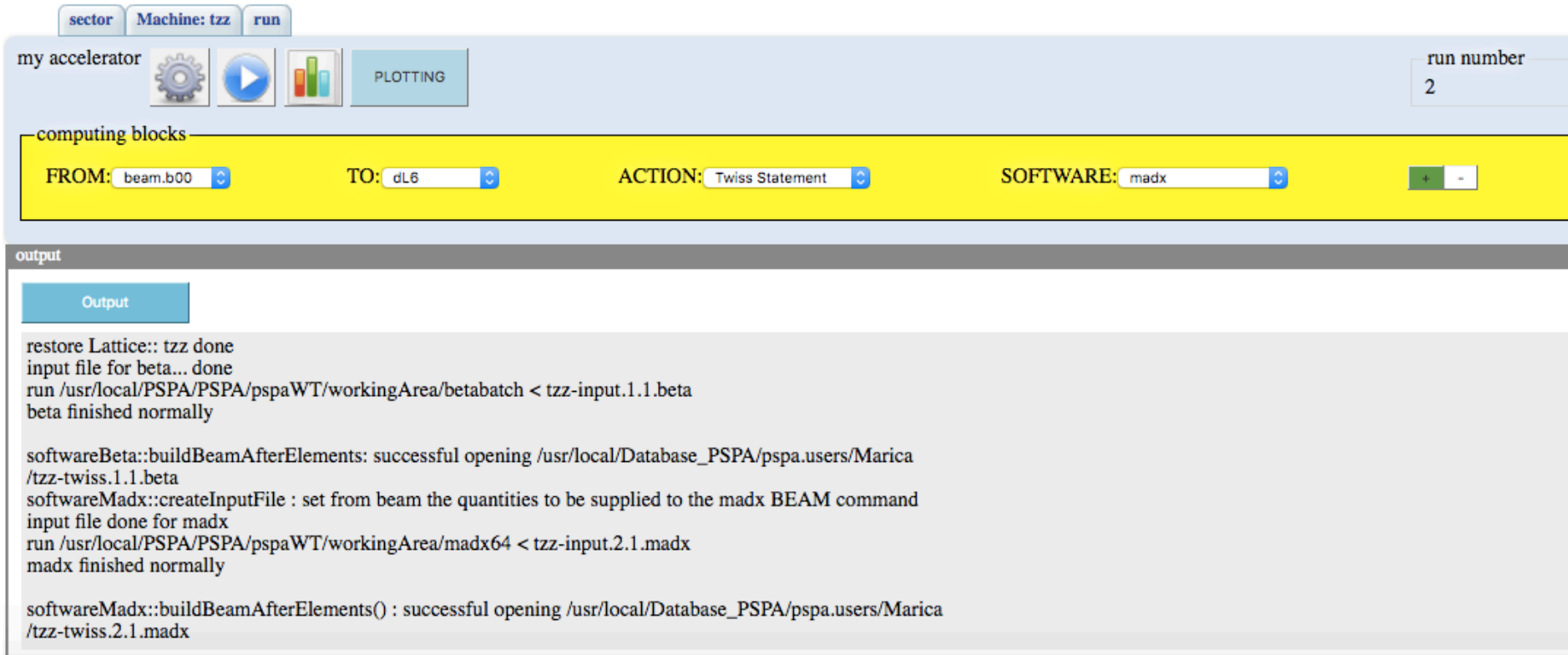
- Select “code”
- Hit RUN

Accelerator elements sidebar containing icons for GUN, BEAM, DRIFT, BEND, RF CELL, SOLEN, FIT, MARK, MULT, QUAD, SEXT, and TR WAVE. Below the sidebar is the SectorList section with the heam logo.

The RUN window interface includes a top toolbar with 'sector', 'Machine: tzz', and 'run' buttons. The main area is titled 'my accelerator' and contains a 'computing blocks' section with fields for 'FROM: beam.b00', 'TO: sept2', 'ACTION: Twiss Statement', and 'SOFTWARE: unknownSoftware'. A dropdown menu for 'SOFTWARE' is open, showing options: 'parmela', 'generator', 'astra', 'transport', 'madx', 'beta', 'elegant', 'usersprogram', 'test', and 'unknownSoftware' (selected). An 'output' button and a status message 'restore Lattice:: tzz done' are also visible.

“RUN” window

- The output window below shows a log from code’s execution



The screenshot shows the PSPA 'RUN' window interface. At the top, there are buttons for 'sector', 'Machine: tzz', and 'run'. Below these, the text 'my accelerator' is displayed next to a gear icon, a play button, a bar chart icon, and a 'PLOTTING' button. On the right side, there is a 'run number' field containing the value '2'. The main configuration area is highlighted in yellow and contains the following settings:

- FROM: beam.b00
- TO: dL6
- ACTION: Twiss Statement
- SOFTWARE: madx

Below the configuration area is an 'output' window with a blue 'Output' button. The output log contains the following text:

```

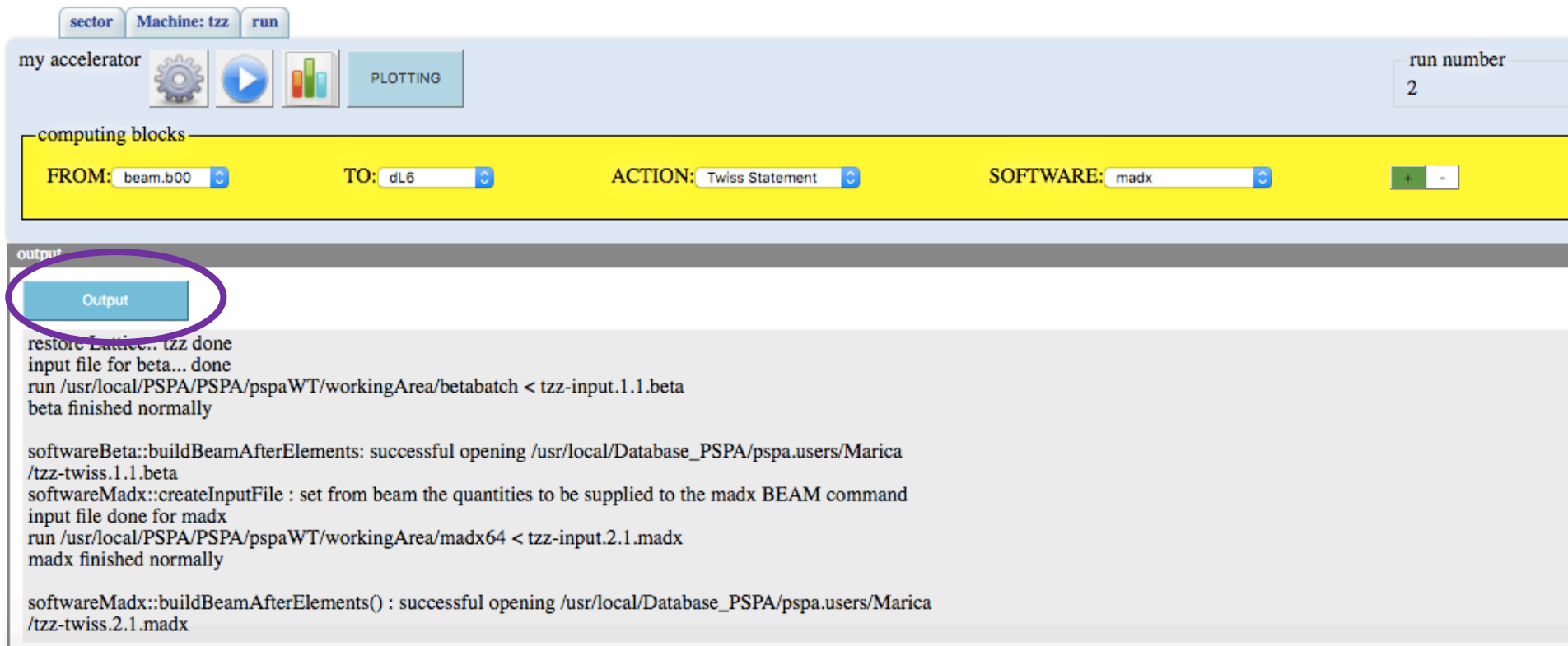
restore Lattice:: tzz done
input file for beta... done
run /usr/local/PSPA/PSPA/pspaWT/workingArea/betabatch < tzz-input.1.1.beta
beta finished normally

softwareBeta::buildBeamAfterElements: successful opening /usr/local/Database_PSPA/pspa.users/Marica
/tzz-twiss.1.1.beta
softwareMadx::createInputFile : set from beam the quantities to be supplied to the madx BEAM command
input file done for madx
run /usr/local/PSPA/PSPA/pspaWT/workingArea/madx64 < tzz-input.2.1.madx
madox finished normally

softwareMadox::buildBeamAfterElements() : successful opening /usr/local/Database_PSPA/pspa.users/Marica
/tzz-twiss.2.1.madox
  
```

“RUN” window

- Hit the Output button to see the code’s output file



The screenshot shows the PSPA 'RUN' window interface. At the top, there are buttons for 'sector', 'Machine: tzz', and 'run'. Below these, there is a 'my accelerator' section with icons for a gear, a play button, a bar chart, and a 'PLOTTING' button. On the right, there is a 'run number' input field with the value '2'. The main area is a yellow bar for 'computing blocks' with fields for 'FROM: beam.b00', 'TO: dL6', 'ACTION: Twiss Statement', and 'SOFTWARE: madx'. Below this is an 'output' section with a blue 'Output' button circled in purple. The output text shows the execution of 'tzz' and 'madx' commands, including file creation and successful completion messages.

“RUN” window

- Example of MADX output file

software Output

Search:

```
+++++  
+ MAD-X 5.01.00 (64 bit, Linux) +  
+ Production Version +  
+ Code Modification Date: 2013.03.13 +  
+ Execution Time Stamp: 12.05.17 13.30.13 +  
+++++  
TITLE,'tzz';
```

```
BEAM, PARTICLE= electron, PC= 0.05, Ex= 0.00015, EY= 0.00015;
```

```
EN:=0.05;
```

```
beam.dL0: DRIFT, L=0.4;
```

```
qp0L: QUADRUPOLE, L=0.15, K1= -1.30602;
```

```
dL2: DRIFT, L=0.2;
```

```
qp1L: QUADRUPOLE, L=0.15, K1= 10.8134;
```

```
qp2L: QUADRUPOLE, L=0.15, K1= -9.60993;
```

Close

Twiss “action”

- Calculation of optical functions β , α , dispersion, tunes
- When action “Twiss” is selected, a window allows for selecting periodical Twiss functions or setting initial values for non periodical solutions
- Initial values by default are taken by the input file, but can be changed at will

ACTION COMMANDS : Twiss Statement

twiss commands

periodic solution
 initial values

Initial values

<input checked="" type="checkbox"/> betx	40.4763
<input checked="" type="checkbox"/> bety	40.9987
<input checked="" type="checkbox"/> alfx	-1.9657
<input checked="" type="checkbox"/> alfy	-1.9325
<input type="checkbox"/> mux	
<input type="checkbox"/> muy	
<input type="checkbox"/> dx	
<input type="checkbox"/> dpx	

Ok Cancel

run number: 0

SOFTWARE: unknownSoftware

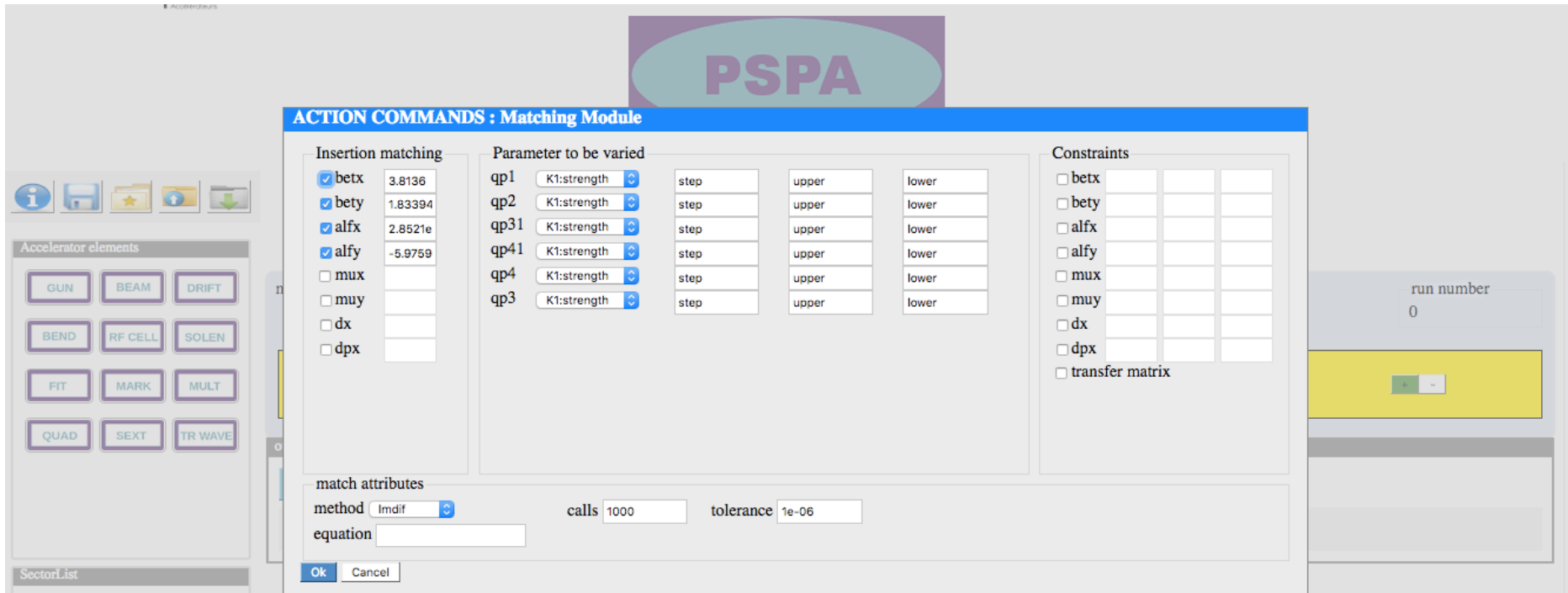
my accelerator Machine: tzz run

computing blocks FROM: beam.b00 TO: sept2

output Output restore Lattice:: tzz done

Match “action”

- Matching of optical functions β , α , dispersion, tunes as well as one-turn matrix elements can be performed
- When action “Match” is selected, a window allows for selecting “constraints” and “variables” for matching and minimization method
- Initial values by default are taken by the input file, but can be changed at will



The screenshot shows the PSPA software interface with the 'ACTION COMMANDS : Matching Module' dialog box open. The dialog box is divided into several sections:

- Insertion matching:** A list of parameters with checkboxes and numerical values:

<input checked="" type="checkbox"/> betx	3.8136
<input checked="" type="checkbox"/> bety	1.83394
<input checked="" type="checkbox"/> alfx	2.8521e
<input checked="" type="checkbox"/> alfy	-5.9759
<input type="checkbox"/> mux	
<input type="checkbox"/> muy	
<input type="checkbox"/> dx	
<input type="checkbox"/> dpx	
- Parameter to be varied:** A table with columns for parameter name, type, and three constraint options (step, upper, lower):

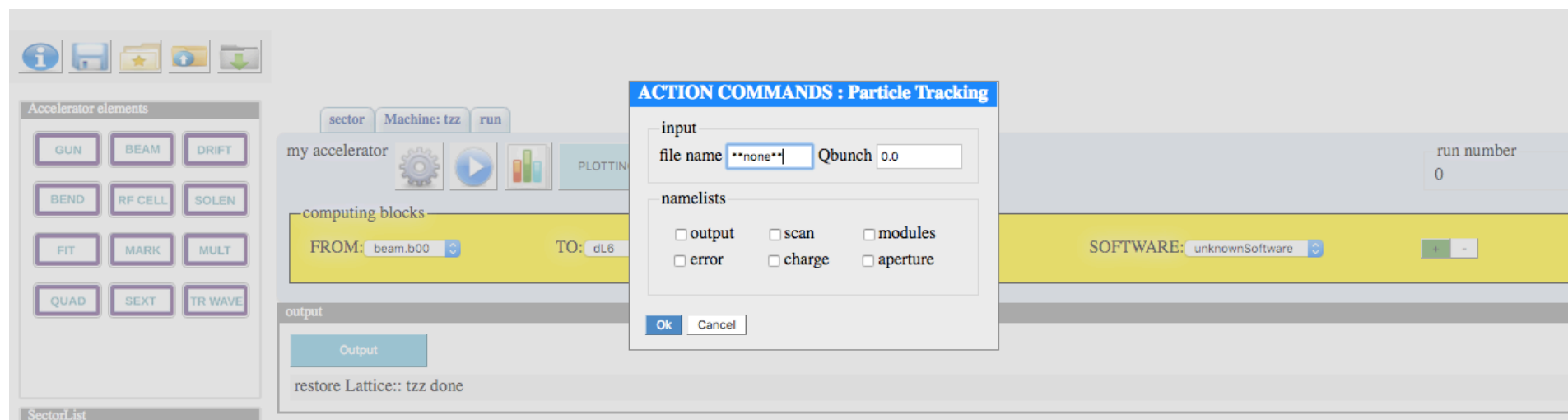
qp1	K1:strength	step	upper	lower
qp2	K1:strength	step	upper	lower
qp31	K1:strength	step	upper	lower
qp41	K1:strength	step	upper	lower
qp4	K1:strength	step	upper	lower
qp3	K1:strength	step	upper	lower
- Constraints:** A grid of checkboxes for various parameters:

<input type="checkbox"/> betx			
<input type="checkbox"/> bety			
<input type="checkbox"/> alfx			
<input type="checkbox"/> alfy			
<input type="checkbox"/> mux			
<input type="checkbox"/> muy			
<input type="checkbox"/> dx			
<input type="checkbox"/> dpx			
<input type="checkbox"/> transfer matrix			
- match attributes:**
 - method:
 - calls:
 - tolerance:
 - equation:

At the bottom of the dialog box are 'Ok' and 'Cancel' buttons. On the right side of the main interface, there is a 'run number' field with the value '0' and a yellow button with a minus sign.

Particle tracking “action”

- Particle tracking command is used to run Parmela@LAL or ASTRA codes
- When action “Particle tracking” is selected, a window allows for selecting the “beam” input file, input the beam charge and select different actions



Particle tracking “action”

- When “output” is selected a start and end location for the lattice is requested
- Zemit is the number of steps for which the emittance value is saved
- Zphase is the number of steps for which a file with particles coordinates and beam properties is saved

The screenshot displays the PSPA software interface. A central dialog box titled "ACTION COMMANDS : Particle Tracking" is open, showing the "input" section with a "namelist OUTPUT" and "parameters" table:

parameters	
ZSTART	0.0
ZSTOP	13.2068
Zemit	100
Zphase	1

The background interface includes a top navigation bar with "sector", "Machine: tzz", and "run" buttons. Below this, there are sections for "my accelerator" (with a gear icon), "computing blocks" (with "FROM: beam.b00" and "TO: dL6" dropdowns), and "output" (with an "Output" button). A "SOFTWARE:" dropdown is set to "unknownSoftware". A "run number" field shows "0". At the bottom, a status bar reads "restore Lattice:: tzz done".

Particle tracking “action”

- When “charge” is selected several parameters are requested (see code’s manuals)

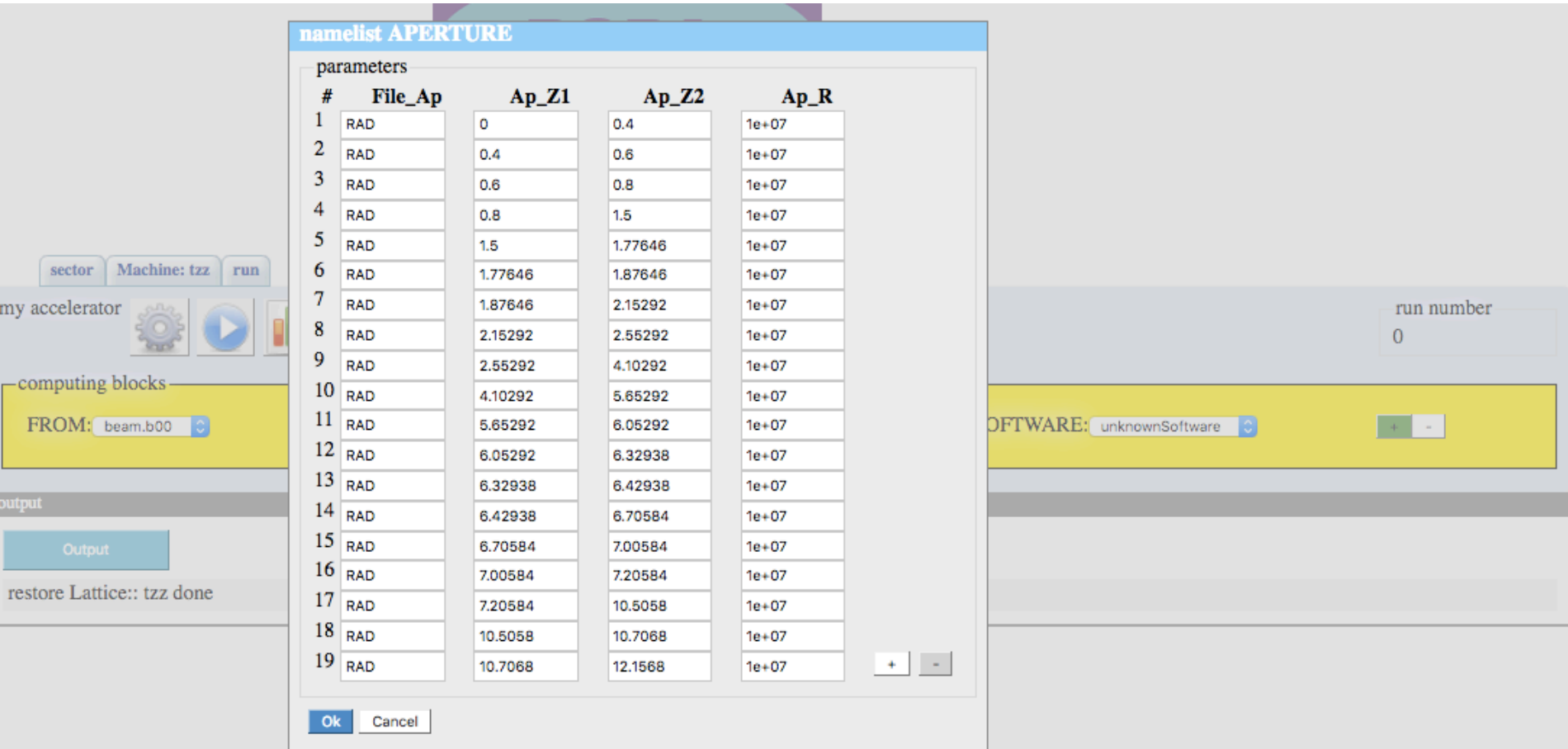
The screenshot displays the PSPA software interface. A central dialog box titled "ACTION COMMANDS : Particle Tracking" is open, showing a "namelist CHARGE" configuration window. The window contains the following parameters for a cylindrical grid algorithm:

Parameter	Value
Nrad	10
N_min	30
Max_scale	0.05
Lmirror	<input checked="" type="checkbox"/>
Cell_var	2.0
min_grid	0.0
Nlong_in	10

The background interface includes a "sector" dropdown, "Machine: tzz", and a "run" button. Below these are "ny accelerator" and "computing blocks" sections. The "FROM:" dropdown is set to "beam.b00". On the right, there is a "run number" input field with the value "0" and a "SOFTWARE:" dropdown set to "unknownSoftware". At the bottom, there is an "Output" button and a status bar that reads "restore Lattice:: tzz done".

Particle tracking “action”

- When “aperture” is selected the beam pipe apertures can be given



The screenshot shows a software interface with a central dialog box titled "namelist APERTURE". The dialog box contains a table with the following data:

#	File_Ap	Ap_Z1	Ap_Z2	Ap_R
1	RAD	0	0.4	1e+07
2	RAD	0.4	0.6	1e+07
3	RAD	0.6	0.8	1e+07
4	RAD	0.8	1.5	1e+07
5	RAD	1.5	1.77646	1e+07
6	RAD	1.77646	1.87646	1e+07
7	RAD	1.87646	2.15292	1e+07
8	RAD	2.15292	2.55292	1e+07
9	RAD	2.55292	4.10292	1e+07
10	RAD	4.10292	5.65292	1e+07
11	RAD	5.65292	6.05292	1e+07
12	RAD	6.05292	6.32938	1e+07
13	RAD	6.32938	6.42938	1e+07
14	RAD	6.42938	6.70584	1e+07
15	RAD	6.70584	7.00584	1e+07
16	RAD	7.00584	7.20584	1e+07
17	RAD	7.20584	10.5058	1e+07
18	RAD	10.5058	10.7068	1e+07
19	RAD	10.7068	12.1568	1e+07

Below the table are "Ok" and "Cancel" buttons. To the right of the table are "+" and "-" buttons. The background interface includes a "sector" dropdown, "Machine: tzz", a "run" button, a "my accelerator" section with a gear icon and a play button, a "computing blocks" section with a "FROM: beam.b00" dropdown, an "output" section with an "Output" button, and a "restore Lattice:: tzz done" message. On the right side, there is a "run number" input field with the value "0" and a "SOFTWARE:" dropdown with "unknownSoftware" selected, along with "+" and "-" buttons.

Particle tracking “action”

- When “error” is selected errors in the lattice (ASTRA code) can be given (see manual)

namelist ERROR

flags		bunch		cavity		solenoid	
Loop	<input type="checkbox"/>	Err_cutoff	3	Err_C_xoff(1)	1E-04	Err_S_xoff(1)	1E-04
LError	<input type="checkbox"/>	Err_Qbunch	5	Err_C_yoff(1)	1E-04	Err_S_yoff(1)	1E-04
ErrorS	<input type="checkbox"/>	Err_Xoff	0.1	Err_phi(1)	1.0	Err_S_xrot(1)	1E-02
Log_Error	<input type="checkbox"/>	Err_Yoff	0.1	Err_MaxE(1)	5	Err_S_yrot(1)	1E-02
LExtend	<input type="checkbox"/>	Err_Toff	3E-04	Err_C_xoff(2)	1E-04	Err_MaxB(1)	5
Suppress_output	<input checked="" type="checkbox"/>	Err_XYrms	10	Err_C_yoff(2)	1E-04		
		Err_Zrms	20	Err_phi(2)	1.0		
				Err_MaxE(2)	5		

Ok

Cancel