02: Introduction to the elegant code

WONG, Chun Yan Jonathan; HAO, Yue;
LUND, Steven; RICHARD, Christopher;

USPAS Accelerator Physics

June 2018

(Version 20180606)

MICHIGAN STATE UNIVERSITY

U.S. DEPARTMENT OF ENERGY | Office of Science

This material is based upon work supported by the U.S. Department of Energy Office of Science under Cooperative Agreement DE-SC0000661, the State of Michigan and Michigan State University. Michigan State University designs and establishes FRIB as a DOE Office of Science National User Facility in support of the mission of the Office of Nuclear Physics.

The elegant Code for Beam Simulation

- Acronym for: ELEctron Generation ANd Tracking
- Developed and maintained by Argonne National Lab (chief architect: Michael Borland)
  - Download: https://www.aps.anl.gov/Accelerator-Operations-Physics/Software#12345
  - Manual: https://ops.aps.anl.gov/manuals/elegant_latest/elegant.html
  - Users forum: https://www3.aps.anl.gov/forums/elegant/
- Freely distributed, multi-platform
- Large code with many options
- Widely used in accelerator community, particularly for e^- machines
  - Storage rings: APS, NSLS, Diamond, SLS, Max IV, etc.
  - FEL driver linacs: LCLS, XFEL, FERMI, SPring-8, etc.
- We will use with “cloud” interface provided by RadiaSoft
- Parallelized: Pelegant

Conventional elegant Runs: an Overview

- Two necessary input files:

  Lattice file (extension ".lte")

  Command file (extension ".ele")

- Running elegant (conventionally via command line)
  - » elegant command_file_name.ele

- Process output files
  - All in SDDS (Self-Describing Data Set) format
  - Analyzed using the SDDS Toolkit: https://www.aps.anl.gov/Accelerator-Operations-Physics/Software#SDDS%20Binaries

Lattice File (*.lte)

- Notation similar to MAD (another popular code)
  - Define elements, then use them to form sequences
  - Comments are preceded by "!"

```
1  ! *** Define Elements ***
2
3  ! Quads
4  Q1: quad,l=0.25,k1=1.0
5  Q2: quad,l=0.25,k1=-1.0
6
7  ! Drifts
8  D0: drift,l=0.5
9  D1: drift,l=2.0
10
11 ! Bend
12 B1: sbend, l=1.0,angle=0.314159265,e1=0,e2=0
13
14
15 ! *** Build Beamline ***
16
17 BL: line=(D0,Q1,D0,Q2,D1,B1,D1)
18 RING: line=(20*BL)
19
```

Define lattice elements, each with specific parameters

Order of definition does not matter

Syntax:

Name: type, parameters / options

Build beamlines from elements

Elements can be other defined lines

Must be listed in order traversed

Syntax:

Name: line=(1st element, …, last element)

Beam Line Elements

- Drifts

D1: DRIFT, l = 1.

Length [m]

Entrance / exit edge angle [mrad]

- Bends

B1: SBEN, angle=0.7854, e1=0.3927, e2=0.3927, l=1.8

Bending angle [mrad]

- Quadrupole

Q1: QUAD, k1=2.8751, l=0.25

Geometric quadrupole strength [m^{-2}]

k1>0: focusing in x
k1<0: defocusing in x

- Sextupole

SD: SEXT, k2=10, l=0.25

Geometric sextupole strength [m^{-3}]

Beam Line Elements

- RF Cavities

CAV: RFCA, l=0.1, volt=3.3e6, freq=5e8, phase = 60

Voltage [V] Frequency [Hz] Phase [degree]

- Marker
  - Allows optimization to be performed at marker position

M1: MARK, fitpoint=1

- Watch
  - output tracking results

Output file name, %s = rootname

W1: WATCH, filename="%s.w1", MODE="coord"

- Aperture
  - To allow particle loss

AP: RCOL, x_max=0.1, y_max=0.15

Command File (*.ele)

```
&run_setup
        lattice = example.lte,
        default_order = 2,
        use_beamline = RING,
        rootname = example,
        final = %s.fin,
        p_central = 107.6
&end

&run_control
&end

&bunched_beam
        n_particles_per_bunch = 10000,
        one_random_bunch=1,
        emit_x = 4.6e-8,
        emit_y = 4.6e-8,
        beta_x = 10, alpha_x = 1,
        beta_y = 10, alpha_y = 1,
        sigma_dp = 0.001,
        sigma_s = 650e-6,
        distribution_type[0] = 3*"gaussian",
        distribution_cutoff[0] = 3*3,
        symmetrize = 1,
        enforce_rms_values[0] = 1,1,1,
&end

&track
&end
```

- Composed of commands addressing different actions
- Each command block
  Starts with “&command_name”
  Ends with “&end”
- Block ordering is important

run_setup

```
&run_setup
        lattice = example.lte,
        default_order = 2,
        use_beamline = RING,
        rootname = example,
        output = %s.out,
        centroid = %s.cen
        p_central = 107.6
&end
```

lattice file name

order of transfer matrices used

line to use

%s means replace by rootname

Final phase space coordinates output to ".out" file

Centroids as function of s output to ".cen" file

$\beta\gamma$ = 107.6

twiss_output

Calculate optics functions

Calculate (matched) periodic solution

```
&twiss_output
        filename = %s.twi,
        matched = 1,
&end
```

Calculate from specified initial conditions

```
&twiss_output
        filename = %s.twi,
        matched = 0,
        beta_x = 10., alpha_x = 1.,
        beta_y = 5., alpha_y = -1.,
        eta_x = 0.4,
        etap_x = 0.,
&end
```

bunched_beam

```
&bunched_beam
    n_particles_per_bunch = 10000,
    emit_x = 4.6e-8,
    emit_y = 4.6e-8,
    beta_x = 10, alpha_x = 1,
    beta_y = 10, alpha_y = 1,
    sigma_dp = 0.001,        sigma_s = 650e-6,
    distribution_type[0] = 3*"gaussian",
    distribution_cutoff[0] = 3*3,
    symmetrize = 1,
    enforce_rms_values[0] = 1,1,1,
&end
```

Gaussian distribution in each plane,

With cutoff at 3 sigmas

Symmetric under change of sign

Distribution must have defined RMS values

Running elegant from Command Line

- To run elegant:
  >>> elegant <filename>.ele

```
LabExercise$ ls
temp.ele  temp.lte
LabExercise$ elegant temp.ele
```

- After the run, many output files appear in the same directory
  - Despite the variety of extensions “.twi”, “.mag”, “.sig”, all are SDDS files
  - It is conventional to write elegant output extensions as acronyms

```
LabExercise$ ls
temp.cen  temp.ele  temp.lte  temp.mag  temp.sig  temp.twi
LabExercise$
```

Output Processing

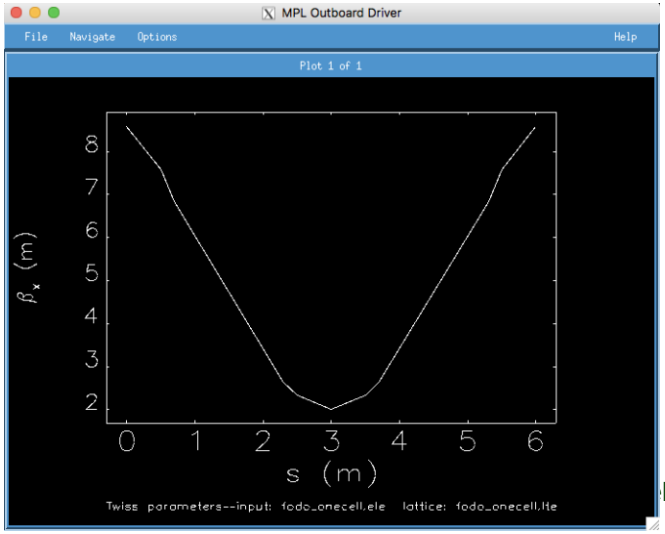
- Employ SDDS Toolkit

>>> sddsprintout –col=s -col=betax –col=alphax –col=betay –col=alphay temp.twi

```
Untitled$ sddsprintout -col=s -col=betax -col=alphax -col=betay -col=alphay temp.twi
Printout for SDDS file temp.twi

       s             betax          alphax          betay          alphay
       m               m                              m
-------------------------------------------------------------------------------
  0.000000e+00   1.160267e+01   2.824466e-17   1.702932e+00  -5.622796e-17
  5.000000e-01   1.023243e+01   2.625325e+00   2.077843e+00  -7.808062e-01
  7.000000e-01   9.213153e+00   2.471064e+00   2.421152e+00  -9.357418e-01
  2.700000e+00   2.377788e+00   9.352187e-01   9.262830e+00  -2.485097e+00
  2.900000e+00   2.035236e+00   7.775399e-01   1.028786e+01  -2.640033e+00
  3.400000e+00   1.661894e+00   1.110223e-16   1.166577e+01   5.551115e-17
  3.900000e+00   2.035236e+00  -7.775399e-01   1.028786e+01   2.640033e+00
  4.100000e+00   2.377788e+00  -9.352187e-01   9.262830e+00   2.485097e+00
  6.100000e+00   9.213153e+00  -2.471064e+00   2.421152e+00   9.357418e-01
  6.300000e+00   1.023243e+01  -2.625325e+00   2.077843e+00   7.808062e-01
  6.800000e+00   1.160267e+01  -2.775558e-16   1.702932e+00  -3.330669e-16
Untitled$
```

>>> sddsplot –col=s, betax temp.twi



Cloud Implementation of elegant Provided Freely by RadiaSoft

- RadiaSoft: http://radiasoft.net/

- Code installed on RadiaSoft servers
  - Access using HTML5 compatible browser interface

- 1) Sirepo
  - https://beta.sirepo.com/#/elegant
  - More user-friendly setup via GUI
  - Post-processing tools readily available

- 2) Python wrapper
  - Implemented using Jupyter notebook
  - Use Python to generate input files and execute runs
  - Post-processing with Python graphics tools

References

- Official website: https://www.aps.anl.gov/Accelerator-Operations-Physics/Software#12345
- Borland, Michael. *Elegant: A flexible SDDS-compliant code for accelerator simulation*. No. LS-287. Argonne National Lab., IL (US), 2000.