ELFIN V77 Users Manual

Takeshi Nakamura
SPring-8
Kamigori-cho, Ako-gun, Hyogo 678-12 JAPAN
nakamura@spring8.or.jp

February 5, 1999

ELFIN is the code which simulates three dimensional time-independent free electron lasers. ELFIN assume input and output are standard input and standard output, respectively. To run ELFIN with 'inputfile' for input and 'outputfile' for output type

```
% ELFIN < 'inputfile' > 'outputfile'
```

1 Structure of Input Files

A input file is composed of three parts. Those are

- Part 1 : Numerical constants.
  The numerical constants used in ELFIN are defined. Those are the number of grid points and the number of superparticles.
- Part 2 : A configuration of an FEL system.
  The configuration of the FEL system is defined. This is the list of names of elements which the component an FEL system, such as amplifiers, mirrors, free spaces for the laser field, etc.
- Part 3 : The definitions of element
  Elements used in part 2 are defined with their parameters.

  For example, if the element name 'FELAMP' is used in part 2 as an FEL amplifier, then in part 3, the element 'FELAMP' should has type 'FEL' which is an FEL amplifier.

  The parameters for the FEL amplifier 'FELAMP', such as length, period, strength of the wiggler, are also given in this part.

The strings written with `typewriting characters` are a part of an input file.
Input files are read with list input `READ(*,*)` in Fortran77.
Lines represented with `COMMENT` appear below are ignored by ELFIN. This line can be used as comment lines.
The structure of input field is two comment lines plus one line for parameters.

2 Part 1 : Numerical constants

The numerical constants are defined. Those are the number of grid points in a transverse space (x, y) and the number of superparticles which are used instead of a huge number of electrons in real FEL systems.

2.1 Format

```
COMMENT
COMMENT
Na   Np
```

2.2 Parameters and Variable Types

`Na` INTEGER
: ELFIN uses `Na`×`Na` square grid points in the transverse space (x, y). `Na` must be 2^n where n is positive integer, which is for FFT, and must not exceed the value **`nam`** which is defined in the file paramsv.inc.

`Np` INTEGER
: The number of superparticles. This value must not exceed the value **`npm`** which is defined in the file paramsv.inc.

If users want to change **`nam`** and **`npm`**, change the values **`nam`** and **`npm`** in the file 'paramsv.inc' and complile ELFIN.

3 Part 2 : Configuration of an FEL system

The configuration of an FEL system is defined.

3.1 Format

```
COMMENT
COMMENT
Nrep
COMMENT
COMMENT
'name1  name2  name3  ... '
```

3.2 Parameters and Variable Type

`Nrep` INTEGER

If there exist `BEGIN` and `END` in the configuration list shown below, ELFIN repeats the elements between them `Nrep` times.

This can be used in a case for oscillators.

`'name1 name2 name3 ...'` CHARACTER*1024

The list of element names which compose an FEL system. The order of the element names is along the propagation of the laser field and the electron beam.

This list is read as single string which surrounded with ′. Each element name is delimited with spaces.

The length of element names must be less than 6 characters.

The last element name in the list must be of the type `TERM`. ELFIN notices that this element is the last element in the list.

4 Part 3 : Definitions of Types and Parameters for Elements

For each element name used in part 2, its type and parameters are defined. The type of the last element in this part must be `TERM`.

The order of the name is not restricted except the type `TERM`.

Each definition of elements has the format below.

4.1 Format

```
COMMENT
COMMENT
'name'  'type'
COMMENT
COMMENT
parameter11 parameter12  ...
COMMENT
COMMENT
parameter21 parameter22  ...
  .
  .
  .
```

4.2 Parameters and Variable Type

`'name'` CHARACTER*6
a name of the element

`'type'` CHARACTER*6
a type of the element

`parameter#` data type is depends on parameter
parameters for the element

5 Types and Parameters of Elements

The types which can be used in part3 are listed below.

`FEL`	A free electron laser amplifier
`ESOURC`	An electron source which creates superparticles
`AINIT`	An initial laser field
`AFREE,ADRIFT`	A free space for a laser field
`MIRROR`	A mirror
`EDRIFT`	A drift space for an electron beam
`TERM`	Terminator of the element list

. The formats for the parameters of these types are mentioned below. ELFIN uses SI units for the parameters.

The widths of calculation areas of the elements which are neighborhood each other must have the same values.

6 Type : FEL

A free electron laser amplifier is simulated.

The output is the laser power, gain and the histogram of the laser field and the electron beam.

6.1 Format

```
================================================================================
<Element name> <Element Type>
   'Name'          'FEL'
--------------------------------------------------------------------------------
<width of the calculation area>
    Width
--------------------------------------------------------------------------------
<# of sampling to z axis> <# of steps to z axis>
    Nsmp                        Nz
---------wiggler parameters-----------------------------------------------------
<Length> <Period>    <Strength>       <transverse dependencies>   <WigglerType>
 Length     lw    (Re(aw0),Im(aw0))      hx2         hy2          WigType
--------------------------------------------------------------------------------
<Wiggler Number> --<Wiggler Parameters (Not used in this version)>--------------
  WigNum             wigpar(1)  wigpar(2)  wigpar(3)  wigpar(4)
-------Spatial filter parameter-------------------------------------------------
<width of filtering> <# of times of splining>
   fltwid                      Nspln
-------- Laser Field  ----------->|<---- Particles ----------------------------
-<histogram>----<list of values>--|--<histogram>--<values>----
  ahswx  ahswy   ameswx ameswy       phswx  phswy  parsw
```

6.2 Parameters

`Width` REAL
: Width of the calculation area $[m]$ in (x, y) space.

`Nsmp` INTEGER
: Sampling step to output. ELFIN outputs every `Nz/Nsmp` step.

`Nz` INTEGER
: The number of step to divide an FEL system along z axis. The wiggler is equally divided to N_z peacies. For positive value of this, the corrector step [**?**] in subroutine feladv for particles and fields is applied twice. For Negative value, the corrector step is applied once.

`Length` REAL
: The length of the wiggler $[m]$.

`lw` REAL
: The Period of the wiggler $[m]$.

`Re(aw0),Im(aw0)` REAL
: The amplitude of a complex dimensionless vector potential of the wiggler $(99.8B_W[T](r.m.s.)\lambda_L[m])$. For an usual wiggler, $Im(aw0) = 0$.

`hx2` REAL
: The coefficient of the x dependence of the wiggler field $[m^{-2}]$. $hx2 = h_x^2$.

`hy2` REAL
: The coefficient of the y dependence of the wiggler field $[m^{-2}]$. $hy2 = h_y^2$.
: $aw(x, y) = aw0\left(1 + \frac{1}{2}h_x^2x^2 + \frac{1}{2}h_y^2y^2\right)$, $h_x^2 + h_y^2 = k_w^2$.

`WigType` CHARACTER*6
: A type of the wiggler For the static field wiggler, it is set to be `'STATIC'` and for an electromagnetic wiggler, it is set to be `'OPTIC'`.

`WigNum` INTEGER
: It is used to specify which subroutine for a wiggler field is to be used in this element. In this version, an usual planer wiggler is assigned to be 1.

`wigpar(1-4)` REAL
: Data for wigglers. Users can use these data in their own subroutine for a wiggler specified in `WigNum`.

`fltwid` REAL
: The width for a spatial filtering $[m]$.

`Nspln` INTEGER
: The order of the splining of the spatial filtering.

`ahswx,ahswy` INTEGER
: Switches to control the output of the histograms of the strength of the laser field along the x axis and the y axis,respectively. 1 is on and 0 is off.

`ameswx,ameswy` INTEGER
: Switches to contorol numerical output of the laser field along the x axis and the y axis , respectively. 1 is on and 0 is off.

`phswx,phswy` INTEGER
: Switches to control the histogram of the distribution of the particles along the x axis and the y axis, respectively. 1 is on and 0 is off.

`parsw` INTEGER
: Switch to controls numerical output of the parameters of the particles. 1 is on and 0 is off.

6.3 Parameters

- `aw0` ,`hx` ,`hy`
  It is assumed that the dimensionless vector potential of the wiggler field has the form of

  $$a_W(x,y,z) = a_{W0}(1 + \frac{1}{2}{h_x}^2x^2 + \frac{1}{2}{h_y}^2y^2)$$

  .

- `fltwid`,`Nspln`
  `fltwid` is the size for the spatial filtering which is applied to the laser field at each step. `Nspln` is the order of splining at the spatial filtering.

  The spatial filtering is off. if `fltwid` is larger than `width` or `Nspln` is 0,

  `WigNum` : user subroutine wiggl# and bexf# is called for the wiggler and the external magnetic field, respectively. # is the number specified by WigNum. For example, if
  t Wignum = 1, then subroutines wiggl1 and bexf1 is called and wiggl1 and bexf1 are preinstalled in ELFIN as a planer and constant field undulator and no exteranl magnetic field.

The harmonic number m is sellected automatically. With the wavelength of the input laser field λ_L, the average gamma of the electron beam γ_0 and the wiggler field strength a_{W0}, m is chosen as the nearest value comparing λ_L and $m\frac{\lambda_w}{2\gamma_0^2}\left(1+|a_{W0}|^2\right)$, $m = 1,3,5,\ldots$.

m is used to select appropriate order of Bessel functions in $[JJ]$.

7 Type : ESOURC

An initial electron beam is simulated.

This module generate superparticles according to the assumed distribution. ELFIN treats small number of the superparticles instead of huge number of the real electrons which interact with the laser field in actual FEL because of the limitation of cpu time and memory.

7.1 Format

```
================================================================================
<Element name>      <Element Type>
   'Name'             'ESOURC'
--------------------------------------------------------------------------------
<Peak Current>
   icurr
--------------------------------------------------------------------------------
<Ave. of Gamma>   <Spread of Gamma>
   gmave0              gmspd0
--------------------------------------------------------------------------------
<Distribution Form>
   gmdst
--------------------------------------------------------------------------------
<Ave. of X> <Spread of X> <Ave. of BetaX> <Spread of BetaX>
   xpave0       xpspd0         bxave0           bxspd0
--------------------------------------------------------------------------------
<Ave. of Y> <Spread of Y> <Ave. of BetaY> <Spread of BetaY>
   ypave0      ypspd0          byave0           byspd0
--------------------------------------------------------------------------------
<Distribution Form>
   trsdst
--------------------------------------------------------------------------------
```

```
<# of points on zeta axis>
   NL
--------------------------------------------------------------------------------
<seed of random number>   <renew the seed or not>
   parnr                        nrupdt
--------------------------------------------------------------------------------
<Factor of spontaneous emission>
   sptfct
```

7.2 Parameters and Variable Type

`icurr` REAL
: The peak current[A] of the electron beam.

`gmave0` REAL
: The average of γ.

`gmspd0` REAL
: The spread of γ.

`gmdst` REAL
: The form of the distribution of γ.

`xpave0` REAL
: The average of x [m].

`xpspd0` REAL
: The spread of y [m].

`bxave0` REAL
: The average of β_x.

`bxspd0` REAL
: The spread of β_x.

`ypave0` REAL
: The average of y [m].

`ypspd0` REAL
: The spread of y [m].

`bxave0` REAL
: The average of β_x.

`bxspd0` REAL
: The spread of β_y.

`trdst` REAL
: The distribution form of (x, β_x, y, β_y)

`NL` INTEGER
: The number of the points in ζ space to place the superparticles.

`parnr` INTEGER
: The seed for random number generations.

`nrupdt` INTEGER
: The switch which specifies if the random series is reset with the seed defined with `parnr`, when this element was called.

sptfct REAL
: Set to be 0 on usual usage.

7.3 Parameters

γ and β_x, β_y are the Lorents factor of the electron beam.

- gmdst, trdst
  They specify the distribution form of γ and (x, β_x, y, β_y), respectively.

  If they are set to 0 then the distribution is aumed to be Gaussian and gmspd0,xpspd0,ypspd0,bxspd0,byspd0 are root mean squares.

  If they are set to be a positive value n, then the distribution form is assumed to be

$$\begin{aligned} f(x, \beta_x, y, \beta_y) dx d\beta_x dy d\beta_y &= N_n \exp(-B^n) dx d\beta_x dy d\beta_y \\ B &= \left(\frac{x - \overline{x}}{\Delta_x}\right)^2 + \left(\frac{\beta_x - \overline{\beta_x}}{\Delta_{\beta_x}}\right)^2 + \left(\frac{y - \overline{y}}{\Delta_y}\right)^2 + \left(\frac{\beta_y - \overline{\beta_y}}{\Delta_{\beta_y}}\right)^2 \end{aligned}$$

  . The values of gmspd0, xpspd0, ypspd0, bxspd0, byspd0 are

$$\Delta_x, \Delta_{\beta_x}, \Delta_y, \Delta_{\beta_y}$$

  .

  If they are set to -1, then the distribution is uniform in 4-dimensional sphere. In this case, gmspd0, xpspd0, ypspd0, bxspd0, byspd0 is the values of the crossing point of the sphere at each axis.

  If they are set to -2, the distribution is uniform in $|x| < x_bound$ and x_bound are gmspd0, xpspd0, ypspd0, bxspd0, byspd0 for each directions.

- gmspd0,xpspd0,ypspd0,bxspd0,byspd0
  The parameters which define the scale of the distribution for each axis, respectively.

- parnr
  The random seed which is used to generate superparticles.

- nrupdt
  If parnr is set to 0 then this element reset the random series with the seed parnr but if parnr is set to 1, then this element does not update the random series and uses the next value of this series.

8 Type : AINIT

An initial laser field is generated.

This element assumes the laser field to be Gaussian(TEM_{00} mode).

8.1 Format

```
================================================================================
<Element name>   <Element type>
    'Name'          'AINIT'
--------------------------------------------------------------------------------
<width of calculation area>
         width
--------------------------------------------------------------------------------
<The wave length>   <The input power>  <waist size>  <point of focus> ---------
         ll                 Pin             w0             lfa
```

8.2 Parameters and Variable Types

`width` REAL
: The width[m] of the calculation area in (x, y) space.

`ll` REAL
: The wave length of the laser field [m].

`Pin` REAL
: The input power of the laser field [W].

`w0` REAL
: The waist size of the laser field when it propagates in free space. This define the width of 1/e point of the laser field [m].

lfa REAL
: The distance to the waist point in free space [m].

9 Type : MIRROR

A reflection of a laser field by a circular spherical mirror is simulated.

9.1 Format

```
==============================================================================
<Element name>   <Element type>
    'Name'          'MIRROR'
------------------------------------------------------------------------------
<width of calculation area>
   width
------------------ Mirror parameter------------------------------------------------
<Diameter> <Curvature> <Reflectance>  ----<Tilt>---- -<Shift of the center> ---
  Diam         Rho          Ref        ThetaX  ThetaY       DeltaX   DeltaY
```

9.2 Parameters and Variable Type

`width` REAL
: The width[m] of the calculation area in (x, y) space.

`Diam` REAL
: The diameter of the mirror [m].

`rho` REAL
: The curvature of the mirror [m].

`ref` REAL
: The reflectance of the mirror.

`ThetaX` REAL
: The x component of the normal vector of the mirror axis.

`ThetaY` REAL
: The y component of the normal vector of the mirror axis.

`DeltaX` REAL
: The shift of the mirror center to x direction [m].

`DeltaY` REAL
: The shift of the mirror center to y direction [m].

10 Type : AFREE

A free space for a laser field is simulated.

10.1 Format

```
==============================================================================
<Element name>  <Element type>
    'Name'         'AFREE'
------------------------------------------------------------------------------
<width at the entrance>   <width at the exit>
     Width_In                  Width_Out
------------------------------------------------------------------------------
<The length of the drift space>
     Length
```

10.2 Parameters and Variable Type

Width_In REAL
: The width of the calculation area at the entrance to the drift space $[m]$.

Width_Out REAL
: The width of the calculation area at the exit from drift space $[m]$.

Length REAL
: The length of the drift space $[m]$.

11 Type : ADRIFT

A drift space for a laser field is simulated. Completely same as type AFREE except differece of names of types.

11.1 Format

```
==============================================================================
<Element name>  <Element type>
    'Name'         'ADRIFT'
------------------------------------------------------------------------------
<width at the entrance>   <width at the exit>
        Width_In                  Width_Out
------------------------------------------------------------------------------
<The length of the drift space>
    Length
```

11.2 Parameters and Variable Type

Width_In REAL
: The width of the calculation area at the entrance to the drift space $[m]$.

Width_Out REAL
: The width of the calculation area at the exit from drift space $[m]$.

Length REAL
: The length of the drift space $[m]$.

12 Type : EDRIFT

A drift space for electrons is simulated.

12.1 Format

```
================================================================================
<Element name>   <Element type>
   'Name'            'EDRIFT'
--------------------------------------------------------------------------------
<The length of drift space>
        Length
```

12.2 Parameters and Variable Type

`Length` REAL
The length of drift space $[m]$.

13 Type : DSPSCT

A dispersive section for bunching like used in optical klystrons.

13.1 Format

```
================================================================================
<Element name> <Element type>
   'Name'         'DSPSCT'
--------------------------------------------------------------------------------
<width at the entrance> <width at the exit>
    Width_In                Width_Out
--------------------------------------------------------------------------------
<The physical length> <effective length>
    Length              Lengthd
```

`Width_In` REAL
The width of the calculation area at the entrance to the dispersive section $[m]$.

`Width_Out` REAL
The width of the calculation area at the exit from the dispersive section $[m]$.

`Length` REAL
The physicsl length of the dispersive section $[m]$.

`LengthD` REAL
The effective length of the dispersive section $[m]$. *LengthD* is related to $\Delta\phi$, the difference of the phase of a laser between the entrance and the exit of the dispersive section. If we use L_d for *LengthD*, $\Delta\phi$ is

$$\begin{aligned}\Delta\phi &= k_L(L_d - \frac{c}{v}L_d) \\ &= -\frac{k_L L_d}{2\gamma^2}\end{aligned}$$

and if we define N_d as

$$\Delta\phi = -2\pi N_d \tag{1}$$

then

$$N_d = \frac{k_L L_d}{4\pi\gamma^2}$$

13.2 Parameters and Variable Type

`Length` REAL
The length of drift space $[m]$.

14 type : TERM

This shows ELFIN that this is the last element of the list of elements which defined in part 2.
This also shows ELFIN that this is the end of part 3.

14.1 Format

```
================================================================================
 <Element name>   <Element type>
   'Name'            'TERM'
```

15 USER ROUTINES

Users can integrate your subroutine to ELFIN. and sample subroutines are pre-installed in ELFIN.
The type of a sample user element is 'new'.
Assume that the element name for a sample element is 'newname'.
The parameters for the element 'newname' are read from file with the soubroutine 'innew' then these data are saved to a buffer for element name 'newname' with subroutine 'ptnew'.
When this element was called in main with call rnnew' with element name 'new', then rnnew call gtnew first to load data from the buffer for element name 'new' then the data was passwd to subroutine new which is the main routine of element type new. new runs with the data for element name 'new'.