

Synrad: Program for Calculating Synchrotron Radiation Power

February 23, 2017

1 Introduction

Synrad is a program for calculating the synchrotron radiation power deposition on the beam chamber walls in a storage ring or accelerator. **Synrad** works by tracking X-rays from creation at the beam to absorption at the chamber wall. The tracking is confined to the horizontal plane. Reflections from the wall are not considered. The vertical extent of the radiation stripe at the wall, important for calculating the power deposition per unit area, is calculated using the vertical emittance.

The chamber wall is specified in the horizontal plane by a set of vertex points with a straight line between points. Exit and entrance lines like x-ray beam lines or dumps can be modeled in **Synrad**. Shadowing of parts of the wall by other parts is taken into account in the calculation.

Even though **Synrad** only tracks photons in the horizontal plane, **Synrad** is able to handle lattices with non-horizontal bends. Radiation from non-horizontal bends is included in the calculation. Obviously, if this radiation is actually striking the top or bottom walls, **Synrad** will not give accurate results.

Synrad output includes power deposition per longitudinal length, per unit area on the beam center line, and photon flux. The beam orbit is not constrained to be zero so calculations to see the affect of varying the orbit are possible.

Synrad is not to be confused with another program called Synrad3D. The Synrad3D program uses Monte Carlo to generate photons along with a full three dimensional chamber model along with a model for the scattering the photons. The advantage of using **Synrad** is that it is quick and accurate since it does not rely on Monte Carlo methods.

Synrad uses the Bmad subroutine library for relativistic charged-particle simulations[?] for some of the calculations.

2 Running Synrad

Syntax for invoking **Synrad**:

```
<path-to-synrad-exe>/synrad {-plot} {<master-input-file-name>}
```

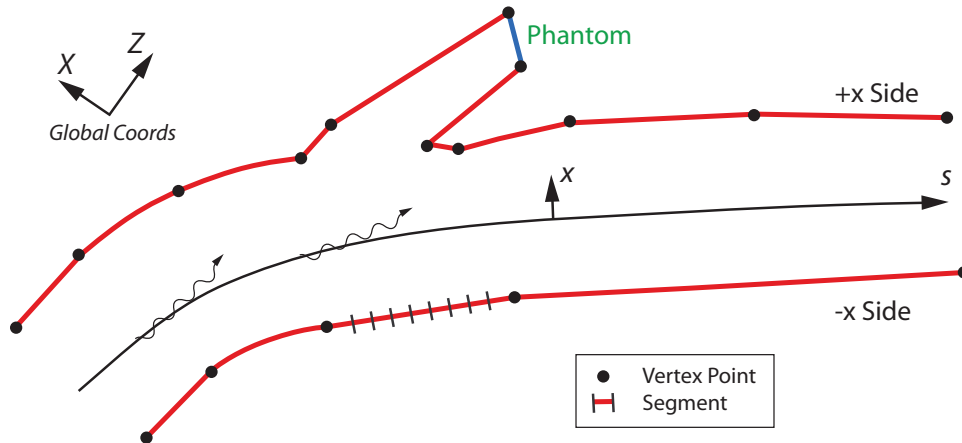


Figure 1: The vacuum chamber wall has a $-x$ side and a $+x$ side. The wall is defined by a number of vertex points (shown as circles) with straight lines in (x, s) between vertices. For the calculation, the straight lines are divided into a number of roughly equal length short segments as shown for a portion of the $-x$ side. Sections between vertex points can also be declared as “phantom” indicating that radiation striking such a section is not to be counted.

The `<master-input-file-name>` optional argument is used to set the master input file name. The default is “`synrad.in`”.

The optional `-plot` option causes `Synrad` to plot the vacuum chamber walls. In this case, no synchrotron radiation calculations are done.

3 Vacuum Chamber Wall

`Synrad` uses the Bmad local reference coordinates for specifying the vacuum chamber wall (see the `Coordinates` chapter in the Bmad manual[?]). The s -axis in this curvilinear coordinate system is the longitudinal axis along the direction of the beam. If there are no vertical bends, the y -axis is in the vertical direction, and the x -axis is in the horizontal plane.

Since `Synrad` is only tracking in the horizontal plane, the vacuum chamber wall is divided into a “positive x ” (“ $+x$ ”) side and a “negative x ” (“ $-x$ ”) side as shown in Fig. ???. The “ $+x$ ” section is defined as the wall on the positive x side of the centerline and vice versa.

The $-x$ and $+x$ sides are specified by a number of vertex points. Between vertex points, a straight line interpolation in (x, s) space is used to construct the wall. Since the coordinate system is curvilinear, the actual shape of the wall in the global coordinate system can be curved. For example, within a bend, a line of constant x in (x, s) space will be mapped to the arc of a circle in the (X, Z) global coordinate system.

Sections between vertex points can also be declared as “phantom” indicating that radiation striking such a section is not to be counted. This is useful for simulating X-ray beam lines

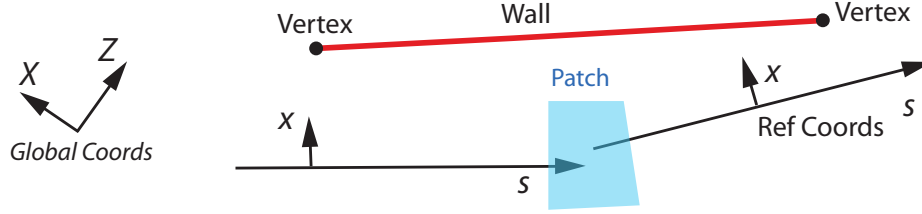


Figure 2: The section of the vacuum chamber wall between two vertices that span a patch element is constructed using a straight line in the global coordinate system.

where the X-rays traveling down the beam line does not contribute to the power dissipated on the walls.

For the calculation, the lines between vertex points are broken up into roughly equal length segments. While **Synrad** tries to make segments of equal size, since the distance between vertex points may not be commensurate with the desired segment length, the actual segment length can vary. For calculational purposes, segments are treated as straight line segments in the global coordinate system. That is, within a bend, only the ends of the segments will truly lie on the straight line, in (x, s) space, drawn between vertices. Since the maximum segment length should be set to be much smaller than any bend radius, the deviation caused in taking the segments as straight in the global coordinate system should be negligible.

Interpolation of the vacuum chamber wall between vertex points that span a **patch** element is problematical in that, in general, an unbroken line in the (x, s) coordinates will be discontinuous in the global coordinates. This comes about since **patch** elements can introduce a discontinuity in the local reference coordinates. Since this is not physical, **Synrad** uses a different rule for interpolating between vertex points that span a **patch**. In such a case, linear interpolation in the global coordinate system is used as shown in Fig. ???. With this convention, the wall may be discontinuous in (x, s) space but this is not a problem since the wall will be continuous in the global coordinate system.

In constructing the vacuum chamber wall, **Synrad** will add vertex points at the boundaries of bend magnets. This is generally transparent to the user except it may result in segment lengths shifting slightly. The x values of the added points are interpolated using the x values of the vertex points to either side, ignoring whether there are any patch elements between the vertex points to either side. That is, the interpolation is done in the curvilinear reference coordinate system.

4 Simulation Technique

Photon generation is based on the standard synchrotron radiation formulas, applicable for dipoles quadrupoles, and wigglers. The radiation is assumed to be incoherent, so **Synrad** cannot treat FEL radiation. Since **Synrad** only deals with the total emitted power (as opposed to power emitted in a given energy bandwidth), **Synrad** can handle undulators.

Photons are generated roughly uniformly along the length of any element where photons are produced. The generated photons are tracked to the vacuum chamber wall. Tracking is confined to be in the horizontal plane. The power P_b per unit angle bend of the particle beam trajectory is[?]

$$P_b (\text{W/radian}) = 14.3 \cdot 10^3 I_b (\text{Amps}) [E_b (\text{eV})]^4 [g(1/\text{m})] \quad (1)$$

where I_b is the beam current, E_b is the beam energy, and $g = 1/R$ is the inverse of the bending radius. The power P_{w1} per unit wall length is then

$$P_{w1} (\text{W/m}) = P_b \frac{\sin \theta_g}{L_p(m)} \quad (2)$$

where θ_g is the grazing angle between the photon and the wall and L_p is the length the photon travels from generation to the wall. The power P_{w2} per unit area on the wall centerline is

$$P_{w2} (\text{W/m}^2) = \frac{P_{w1}}{2 \pi \sigma_y} \quad (3)$$

where σ_y is the effective vertical sigma of the photon distribution at the wall

$$\sigma_y^2 = [\epsilon_y \beta_y] + [(\epsilon_y \gamma_y + 1/\gamma^2) L_p^2] \quad (4)$$

where β_y and γ_y are Twiss parameters, ϵ_y is the vertical beam emittance, and γ is the usual beam relativistic factor. The first term in square brackets on the RHS of the equation is the particle beam height and the second term is due to the finite vertical opening angle of the photons.

Note: If the lattice is circular, photons leaving one end of the lattice will be tracked through from the other end. That is, such photons will be handled correctly. If the lattice is not circular, photons reaching the ends of the lattice will be absorbed by the wall there.

For a given element where radiation is produced, a set of N photons are generated and tracked. For wiggler radiation, the number of photons produced per period of beam oscillation must be of order 10 or more for an accurate calculation. Each pair $(i, i+1)$ of photons, with $i = 1, \dots, N-1$, defines a section of the beam trajectory over which radiation is produced. The i^{th} photon strikes the wall at wall segment W_i . For each radiation section, the power incident on the wall segments between W_i and W_{i+1} is computed using interpolation. It is assumed that the segment length is small enough so that power on a segment is

$$P_s = l_s P_{s0} \quad (5)$$

where P_s is the integrated power over the segment, l_s is the segment length, and P_{s0} is the power density at the center of the segment. A check is made to see if any segments are shielded from the photon flux by other parts of the wall. If so, the power on these segments due to the radiation section is set to zero.

4.1 Fortran Namelist

Fortran namelist syntax is used for parameter input by **Synrad**. The general form of a namelist is

```

&<namelist_name>
  <var1> = ...
  <var2> = ...
  ...
/

```

The tag "&<namelist_name>" starts the namelist where <namelist_name> is the name of the namelist. The namelist ends with the slash "/" tag. Anything outside of this is ignored. Within the namelist, anything after an exclamation mark "!" is ignored including the exclamation mark. <var1>, <var2>, etc. are variable names. Example:

```

&section_def section = 0.0, "arc_std", "elliptical", 0.045, 0.025 /

```

here `section_def` is the namelist name and `section` is a variable name. Notice that here `section` is a "structure" which has five components.

5 Main Input File

The main input file can be specified on the command line invoking Synrad3D. If not given, the default name for the main input file is "synrad.init".

Fortran namelist syntax is used (§??). Example main input file:

```

&synrad_params
  sr_param%lat_file = "lat.bmad"      ! Input lattice.
  sr_param%i_beam = 0.1               ! Single-beam current.
  sr_param%epsilon_y = 10e-12         ! Vertical emittance.
  sr_param%n_slice = 20               ! # of slices per element or wiggler period
  sr_param%filter_phantom_photons = T ! Filter photons striking phantom wall?
  seg_len = 0.1                      ! Segment length for calculation.
  beam_direction = 0                 ! -1 = track backwards only,
                                     ! 0 = track both directions, 1 = forward.
  wall_file = "wall.dat"             ! Name of file specifying the wall.
  forward_beam = "POSITRON"          ! "POSITRON" or "ELECTRON"
  backward_beam = "ELECTRON"         ! This is important if there are elsep elements.
  use_ele_ix = 0                     ! If non-zero only emit photons from one element.
/

```

`sr_param%lat_file`

The `lat_file_name` parameter specifies the lattice file to be used. Lattices are in Bmad standard format [?].

`sr_param%i_beam`

The `i_beam` parameter specifies the current (Amps) in a single beam.

`sr_param%epsilon_y`

The `epsilon_y` parameter specifies the vertical emittance to use in the simulation.

`sr_param%n_slice`

The `n_slice` parameter specifies the number of slices to divide magnetic elements into. More slices gives a more accurate depiction of the radiation but increases the computation time and file sizes. For periodic type wigglers `n_slice` is taken to be the number of slices per period. For map type wiggler the number of slices is taken to be equal to the value of the `num_steps` parameter.

`sr_param%seg_len`

The `seg_len` parameter sets the maximum size of a wall segment. A smaller size can give better power deposition resolution but increases the computation time. and the output file size.

`sr_param%filter_phantom_photons`

If `%filter_phantom_photons` is set to `True` (the default), photons hitting a part of the chamber wall that has been designated as `phantom` (§??) are ignored and do not contribute to the output.

`beam_direction`

The `beam_direction` parameter specifies the direction of the tracking to use: -1 = track backwards only (This corresponds to e^- in CESR), 1 = track forward only (e^+ in CESR), and 0 = track both directions.

`wall_file`

The `wall_file` parameter specifies a file giving a list of vertices for the $-x$ and $+x$ sides to define the horizontal aperture of the wall. See below for more details. Synrad3D wall files may also be used. When using a Synrad3D wall file the file name must be prefixed with "synrad3d:". Example:

```
&synrad_params
...
wall_file = 'synrad3d:/home/critten/dc03.wall3d'
/
```

`forward_beam`

The `forward_beam` parameter specifies which species moves in the positive s direction. Should be "POSITRON" for CESR. This parameter only matters if there are `elsep` elements in the lattice.

`backward_beam`

The `backward_beam` parameter specifies which species moves in the negative s direction. Should be "ELECTRON" for CESR. This parameter only matters if there are `elsep` elements in the lattice.

`use_ele_ix`

The `use_ele_ix` parameter specifies the index a single element from which to generate synchrotron radiation. If this is 0, then generate power from all bends, quads and wigglers.

6 Vacuum Chamber Wall Definition

Wall files define a series of wall vertices. The definition includes the s -position (meters), and the corresponding x values for the $-x$ and $+x$ wall sides. Note: The s -position of the last vertex will be automatically set to the s value at the end of the lattice.

Fortran namelist syntax is used (§??). A single namelist is used to define the wall at a particular s -position. The general form is

```
&wall_pt  s = <val> {x_minus = <val>} {x_plus = <val>}  
                                     {phantom = <T/F>} {name = <string>} /
```

x_minus defines the x value for the $-x$ wall and x_plus defines the value for the $+x$ wall.

Example:

```
&wall_pt  s =    0.0000  x_minus =  -0.0450  x_plus =    0.0450  name = BP1 /  
&wall_pt  s =    0.8581  x_minus =  -0.0490  x_plus =    0.0490  name = WIG1 /  
&wall_pt  s =    0.8581                               x_plus =    0.1223  phantom = T /  
...
```

Note: An old notation uses x_in for x_minus and x_out for x_plus .

Either x_minus , or x_plus , or both must be present in each `wall_pt` namelist. If both are present, two vertex points are being defined at once.

The `name` parameter is used for identifying individual vertex points.

The wall is constructed by connecting each vertex point to the previous point. Phantom wall sections can be specified setting `phantom` to T for a vertex. In this case the wall section between the vertex and the previous vertex will be marked a phantom.

7 Output Files

There are five output files generated:

```
element_power.dat  
synch_power_negative_x_side.dat  
synch_power_positive_x_side.dat  
synrad_negative_x_side.txt  
synrad_positive_x_side.txt
```

`element_power.dat`

List of all elements where radiation is produced showing the power radiated and the power that hit the walls. These two numbers should be the same.

`synch_power_negative_x_side.dat`, `synch_power_positive_x_side.dat`

List of all wall segments showing such things as power deposited, power per unit length, photons per second impinging, etc.

`synrad_negative_x_side.txt`, `synrad_positive_x_side.txt`

Similar to the `synch_power*.dat` files.

References

[Bmad] D. Sagan, "Bmad: A Relativistic Charged Particle Simulation Library" Nuc. Instrum. & Methods Phys. Res. A, **558**, pp 356-59 (2006).

The Bmad Manual can be obtained at:

<http://www.lepp.cornell.edu/~dcs/bmad>

[Sands] Matthew Sands, *The Physics of Electron Storage Rings, An Introduction*, SLAC-121 Addendum, 1970.