ISBN: 978-3-95450-208-0 doi:10.18429/JACoW-IPAC2019-TUPMP037

RECENT DEVELOPMENTS OF MONTE-CARLO CODES MOLFLOW+ AND SYNRAD+

R. Kersevan[†], M. Ady, CERN, Geneva, Switzerland

Abstract

Molflow+ and Synrad+ are Monte Carlo simulation tools for ultra-high vacuum and synchrotron radiation, respectively. Over the years they have become a common tool for designing and analysing the vacuum system of particle accelerators. This contribution gives a short summary about new features added since the last IPAC contribution [1]. Synrad+ now supports low flux mode, a weighted Monte Carlo technique where the represented number of photons is reduced at every reflection, providing significantly better statistics at low flux regions. As for Molflow+, angle maps allow recording the molecules' directional distribution at any point, and then desorb a reduced gas quantity according to the recording. In linear systems, this allows iterative simulations that have been proven to treat systems up to 7 orders of magnitude of pressure difference. Without the new technique the computing time would be prohibitively slow on desktop computers, which is what most users of the two codes use.

Both codes now have a built-in geometry builder that allows creating simple models through a set of 3D operations and modifying those imported from CAD tools.

Molflow+ has been extended with additional diagnostic tools, such as a logger that records properties of all hits on a scoring surface, and histogram plotters that visualize the distribution of the number of bounces, the distance to absorption and the time of flight of the gas molecules. The codes have recently become open source, and it has been made compatible with, and tested on different versions of Linux and macOS.

CODE OVERVIEW

Molflow+ is a simulator for ultra-high vacuum that has been written in the 1990s and ported to modern C++ based code in 2007 [2]. It uses the test-particle Monte Carlo method, tracing the trajectory of virtual gas molecules from source (a gas injection or thermal outgassing location) to absorption (typically a vacuum pump). The geometry is represented as vacuum boundaries (walls) with polygons. These polygons, extended with physical properties (temperature, sticking and outgassing, sojourn time, etc.) are referred to as *facets*.

As the simulation is running, several counters record hits that belong either to entire facets, or cells of post-processing entities called *textures* or *profiles*. Since the number of counters is defined before the simulation is launched, the memory requirement remains the same throughout a run. Using these counters, physical quantities such as pressure, density and impingement rates are calculated and updated on the screen every second. The calculated values and the color-coded textures and profile plots fluctuate with each screen update, but as the statistical error of test-particle MC simulations decreases with the square root of the number of hits [3], they converge to the solution over time. This allows the user to decide when the results are accurate enough to stop the run, at which point they are visualized internally or exported for further post-processing.

Synrad+ is a code forked from Molflow+, using the same ray-tracing engine. Instead of gas molecules, it traces photons originating from magnetic accelerator elements. These elements, typically dipoles, quadrupoles or periodic elements like wigglers, are referred to as *magnetic regions*. With user input defining beam properties, starting point position and direction, they are represented as a number of trajectory points, each of which can generate a virtual photon representing a certain photon flux. These virtual photons are then traced through the geometry, hitting wall facets. Upon a hit, reflection, absorption, and optionally backscattering and transmission probabilities depend on the wall material, roughness, incident angle and photon energy. Such probability tables are included for a few metals and can be defined for new materials by the user.

The two codes share file formats and their interface is similar. A coupled usage of the two codes would be first simulating flux absorption with Synrad+, then converting it to dynamic outgassing in Molflow+ and finally proceeding with a vacuum simulation, as demonstrated in [4].

NEW TOOLS FOR THE GEOMETRY

Earlier versions of the codes imported the geometry through the STL file format, which is extensively supported by CAD programs due to its popularity in 3D printing. That format describes solid bodies' surfaces by a list of triangles, which Molflow+ merges by detecting coplanar and adjacent triangles, colinear sides and shared triangle vertices. Nevertheless, simplifying a real 3D model to a vacuum geometry is a non-trivial process: ideally an experienced mechanical engineer removes non-relevant parts (screws, flanges, mechanical supports), inverts the volume by converting voids in the structure to solid parts, then simplifies curved parts (which can only be described by a large number of planar facets) and joins walls to prevent leaks. Even if done correctly, many CAD programs introduce small rounding errors during the conversion to STL format, and the meshing of surfaces to triangles is arbitrary, making Molflow+ post-processing (orienting textures, etc.) difficult.

Geometry Editor

Due to the issues above, and since many users of the codes are physicist without access to professional CAD tools, a geometry editor was added to the codes, that allows

[†] Corresponding author: roberto.kersevan@cern.ch

modifying imported geometries but also creating new ones from scratch. While it cannot compete with commercial 3D modelling tools, it offers a set of sketching, extrusion and Boolean operations, allowing quick creation of simple vacuum systems (typically a network of cylindrical tubes, see Fig. 1). One particularity of this editor is that it works exclusively with surfaces represented as polygons, as opposed to the usual volume-based CAD logic. One advantage of using the built-in editor is that the resulting facets' orientation will follow the geometry, allowing easier meshing for post-processing.

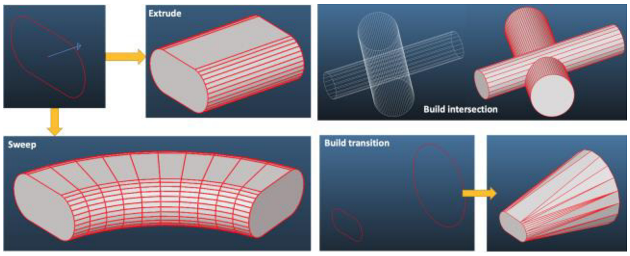


Figure 1: Geometry editor operations.

Generating Geometries for Synrad+

As for Synrad+ geometries, a new tool called *OpticsBuilder* has been published. It bridges a gap between two styles of accelerator description:

- Accelerator physics community treats a machine as a sequence of magnetic elements and drifts, each having a 2-dimensional aperture. It means that the geometry is essentially referenced by the beam's s coordinate, and the actual 3D positions are calculated from the element lengths and the orientation (referenced by the theta angle). Within CERN, the reference code using this description is MADX [5], but it is also adopted by other synchrotron radiation simulators, such as Synrad3D developed at Cornell [6].
- In our codes, following the engineering description, the geometry is referenced by global X, Y, Z coordinates, and the magnetic regions are independent of each other: their position and orientation is defined by the user, and even the beam parameters are unique to each region.

The latter description is more flexible (it allows to include fine details such as crotch absorbers, grids, beamscreens, and can treat multiple beams within one simulation). However, when simulating larger machine parts (such as entire sectors), it requires manual calculation of the global coordinates, and lattice changes are difficult. Therefore, a new tool, written in C# and having a graphical user interface, intends to bridge the gap between the two approaches. It helps generating Synrad+ geometries by the following principles:

- A sequence of elements, either magnetic or drifts, can be defined manually, or imported from a MADX exported Twiss file.
- Each element has a length, parametrically described 2D aperture, and in case of dipoles, a bending angle. One element (for example the beginning of the sequence or the interaction point) serves as a reference, having a global X,Y,Z coordinate and theta orientation.
- Global coordinates are calculated from the reference element's position and subsequent elements' lengths and bends, assuming continuity (no gaps between elements)
- Beam parameters are global throughout the sequence

The geometry, and the magnetic regions (as on Fig. 2) are then generated for Synrad+. One remaining step is to define physical properties (wall materials and roughness) and optionally add post-processing textures before launching the simulation.

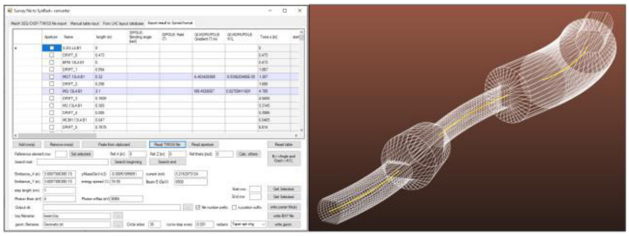


Figure 2: OpticsBuilder interface (left) and generated geometry (right).

SPEEDUP BY IMPORTANCE SAMPLING

In traditional MC simulations, one test particle represents a given, fixed number of molecule or photon flux. One weakness of the approach is that the statistical error of the results within the same geometry will be larger at areas where the pressure or the photon flux is lower. For example, take a vacuum system where gas is injected through a nozzle at 10^{-3} mbar. Through expansion to ultra-high vacuum, the pressure is ultimately reduced to 10^{-9} mbar. Because the pressure spans over 6 orders of magnitude, one million test particles need to be traced in the high-pressure nozzle to get a single hit in the low-pressure part. This either results in low statistics in the low-pressure part or excessively long simulation time. As for synchrotron radiation simulations, the equivalent problem is an accelerator with a low-reflectivity surface: at 1% reflection probability, after 3 reflections the photon flux (and the available statistics) is also reduced by 6 orders of magnitude.

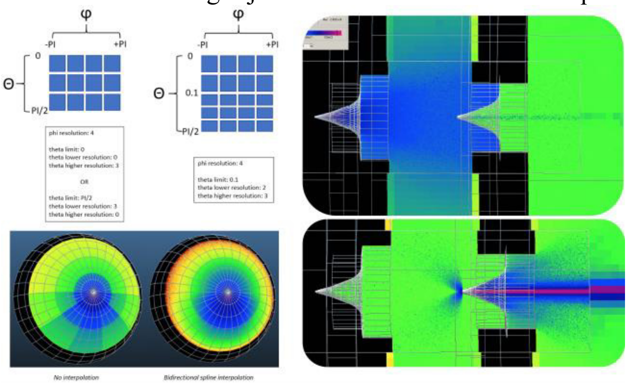
Angle Maps in Molflow+

In case of the nozzle example above, or with any *linear* geometry (which can be partitioned to subsequent parts as the gas flows), one can drastically speed up computation by simulating *in series*:

- we create logical boundaries between the parts
- simulating from the first part, we sample the amount of gas making it to the next volume
- then we relaunch a new simulation with only the passing gas flow generated from the boundary, towards the next volume
- we repeat the steps above until we reach the lowest pressure volume

One issue, however, is that the angular distribution of the generated gas must match the sampled distribution of the previous simulation step. Since linear systems tend to exhibit *beaming*, we cannot assume the isotropic, Lambertian distribution of vacuum systems in equilibrium.

Molflow+ has a new feature called *angle maps*: the user can sample a 2-dimensional angular distribution (elevation and azimuth angles) of the impinging gas jet, then generate gas according to the sampled distribution. Figure 3 shows the simulation of a gas jet with and without the technique.



phi resolution: 4

theta limit: 0
theta lower resolution: 0
theta higher resolution: 3

OR

theta limit: PI/2
theta lower resolution: 3
theta higher resolution: 0

phi resolution: 4

theta limit: 0.1
theta lower resolution: 2
theta higher resolution: 3

Figure 3: angle map description (top left), interpolation effect (bottom left), directed gas jet passing through hole without (top right) and with (bottom right) angle maps.

Low-flux Mode in Synrad+

In synchrotron radiation simulations, geometry partitioning is not necessary. If the *low-flux* mode activated, every time a virtual photon hits a surface with reflectivity R, it is always reflected, however the represented photon flux is reduced to R part of that before the reflection. Due to predominantly specular reflections, the test photons with the guaranteed reflection can travel large distances down the accelerator, reaching low-flux areas. They are eliminated when the ratio of their represented and original flux falls below the *cutoff limit*. As a thumb rule, if we intend to simulate flux across N orders of magnitude, the cutoff limit should be 10^{-N}. The speedup is illustrated in Figure 4.

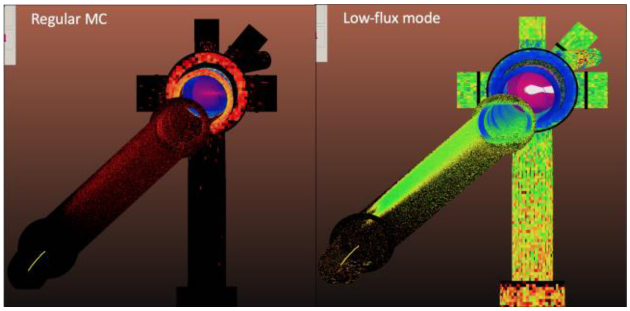


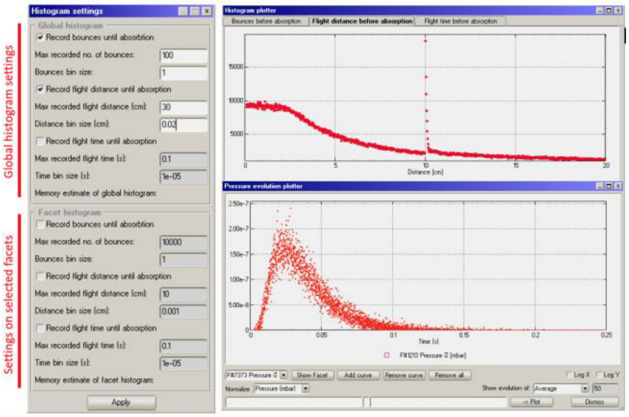
Figure 4: Synrad+ flux maps with low-flux mode disabled (left) and enabled (right) after the same computing time.

ADDITIONAL CHANGES

New Post-processing Tools

On user request, both codes now include a *Particle logger*, that allows to sample a large set of test particles on a scoring surface. The list includes hit positions, directions, and physical properties (time, velocity for molecules and energy, represented flux and power for photons).

Molflow+ now includes a *histogram plotter*, illustrated in Fig. 5, that can sample the number of bounces, flight time and flight distance before absorption for either the whole system or for a particular facet.



Histogram settings

Global histogram

☑ Record bounces until absorbtion
Max recorded no. of bounces: 100
Bounces bin size: 1
☑ Record flight distance until absorption
Max recorded flight distance [cm]: 30
Distance bin size [cm]: 0.02
□ Record flight time until absorption
Max recorded flight time [s]: 0.1
Time bin size [s]: 1e-05
Memory estimate of global histogram:

Facet histogram

□ Record bounces until absorbtion
Max recorded no. of bounces: 10000
Bounces bin size: 1
□ Record flight distance until absorption
Max recorded flight distance [cm]: 10
Distance bin size [cm]: 0.001
□ Record flight time until absorption
Max recorded flight time [s]: 0.1
Time bin size [s]: 1e-05
Memory estimate of facet histogram:

F#7373 Pressure
□ Log X □ Log Y
Normalize: Pressure [mbar]
Show evolution of: Average 50

Figure 5: histogram plotter parameters (left), flight distance histogram in a 10 cm long tube (top right) and flight time histogram following a pulsed gas injection (bottom right).

Cross-platform and Open-source

Molflow+ has been used for simulating contamination processes in satellites, and recently a certain number of collaborations have been set up between CERN and space industry players. As part of this shared effort, both codes are now published under the GNU GPL v2 or later license, with the source code available on the website [7].

Molflow+ has been ported to most desktop platforms: apart from the original Windows version, macOS and Linux variants are now also available. Porting of Synrad+ is under way. The ultimate goal is to allow executing the codes on high-performance computing clusters, predominantly running Linux.

REFERENCES

[1] M. Ady and R. Kersevan, “Introduction to the Latest Version of the Test-particle Monte Carlo Code Molflow+”, in *Proc. 5th Int. Particle Accelerator Conf. (IPAC'14)*, Dresden, Germany, Jun. 2014, pp. 2348-2350. doi:10.18429/JACoW-IPAC2014-WEPME038

[2] Kersevan, Roberto & Pons, J.-L. (2009). Introduction to MOLFLOW+: New graphical processing unit-based Monte Carlo code for simulating molecular flows and for calculating angular coefficients in the compute unified device architecture environment. Journal of Vacuum Science & Technology A: Vacuum, Surfaces, and Films. 27. 1017 - 1023. doi:10.1116/1.3153280.

[3] Y. Suetsugu. Application of the monte carlo method to pressure calculation. Journal of Vacuum Science & Technology A, 14(1):245–250, 1996. DOI: 10.1116/1.579927

[4] M. Ady, R. Kersevan, and M. J. Grabski, “Monte Carlo Simulations of Synchrotron Radiation and Vacuum Performance of the Max IV Light Source”, in *Proc. 5th Int. Particle Accelerator Conf. (IPAC'14)*, Dresden, Germany, Jun. 2014, pp. 2344-2347. doi:10.18429/JACoW-IPAC2014-WEPME037

[5] MADX software, http://mad.web.cern.ch/mad/

[6] Dugan, G, Sagan, D, "SYNRAD3D photon propagation and scattering simulations", in Joint INFN-CERN-EuCARD-AccNet Workshop on Electron-Cloud Effects, La Biodola, Isola d'Elba, Italy, 5 - 9 Jun 2012, pp.117-129 (CERN-2013-002), doi: 10.5170/CERN-2013-002.117

[7] Molflow+ website, https://molflow.web.cern.ch/