# Beamlet Example Documentation

Revised September 27, 2021

# Contents

**1 Getting Started** **5**

1.1 Introduction . . . . . 5

1.2 Setup . . . . . 5

1.3 Running the Beamlets Example . . . . . 6

1.4 Visualization Utilities . . . . . 7

**2 Beamlet Distribution Simulation** **9**

2.1 Boltzmann Velocity Selection . . . . . 9

2.2 Converting Velocity to Scaled Momentum . . . . . 11

**3 Ultracold Beamlet Dynamics** **12**

# List of Figures

2.1 Boltzmann distribution velocity (black) compared to 8000 values sampled in 3D via the Box-Muller transformation. The distribution peak is shown by the red line. . . . . 11

3.1 Electron temperature (K) above and charge density ($\mu$C/m$^3$) below at time $t = 0$ ns, 1.3 ns, 2.0 ns, 2.7 ns . . . . . 13

# List of Data Files

2.1.1 Box-Muller algorithm for generating a 3D Boltzmann distribution. Written for COSY INFINITY. . . . . 10

# Chapter 1

# Getting Started

## 1.1 Introduction

The `make_beamlets_example.fox` file is written in COSYScript for MSU COSY Infinity v9.2 and higher. It generates positions and momentum for particles sampled from a 2D gaussian in the $xy$-plane and uniform within the $z$-plane and whose momentum are sampled from a 3d Maxwell-Boltzmann distribution. It also creates the requisite input files for use with PHAD (PHAD Documentation). User specified variables are given in Tables 1.1, 1.2.

## 1.2 Setup

You must install MSU COSY Infinity to use PHAD and to run `make_beamlets_example.fox`. Contact the beam physics group at the COSY website for permission and instructions. PHAD does require the COSY beam physics package (cosy.fox) to run.

After COSY is installed, you will need some things to generate the example:

1. `make_beamlets_example.fox`
2. `cosy.bin` file (compiled by running `cosy.fox`)
3. `input` (Directory where PHAD input files are written)
4. `output` (Directory where PHAD output files are written)
5. Requisite files/directories for running PHAD (see the PHAD Documentation)
    - Note: `phad-mpi.input` and input particle data required for PHAD are generated by `make_beamlets_example.fox`

| Variable Name | Description |
|---|---|
| ATOMIC_MASS | Atomic mass of each particle ($f_i = m_i/m_p$). |
| TEMPERATURE_IN_KELVIN | Temperature of particle distribution in Kelvin. |
| TEMPERATURE_IN_KELVIN_HALO | Temperature of halo particles in Kelvin. |
| NUM_OUTER_BEAM | Number of outer beamlets. |
| INNER_BEAM | Inner beamlet flag 1 for beamlet or 0 for no inner beamlet. |
| R | Outer radius of beamlet centers. |
| N_OUTER_BEAM | Number of particles in outer beamlets. |
| N_INNER_BEAM | Number of particles in center beamlet. |
| R2 | Beamlet Radius. |
| SIGMA | Specify standard deviation of x,y spacial distribution. |
| ZSCALE | Specify width of uniform distribution of z spacial distribution with mean 0. |
| CHARGE | Charge of each particle. |
| HALO_R | Radius of the uniform halo distribution. |
| N_HALO | Number of halo particles. |

Table 1.1: User inputs for the beamlet distribution in `make_beamlets_example.fox`

There is a section in `make_beamlets_example.fox` for user specified variables running from *lines 418-466*. The first set of variables set the parameters that determine the beamlet distribution and are detailed in Table 1.1. These variables allow the user to set the particle species (mass and charge), number of beamlets and particles per beamlet, and beamlet spatial distribution. In particular, the inclusion of a uniform or Gaussian (uncomment *lines 268-298* and comment *lines 299-300* for Gaussian) distribution of halo particles can be specified. The second set of variables, detailed in Table 1.2 assign the input parameters for PHAD itself, which are used to generate the input file `phad-mpi.input`. More details on these parameters can be found in the PHAD Documentation or application publications at the Beam Physics Code Repository.

## 1.3 Running the Beamlets Example

Assuming the COSY executable is `'cosy'`, the basic command is

```
cosy make_beamlets_example.fox
```

This will generate the PHAD input files

```
positions.ssv
masses.ssv
momenta.ssv
charges.ssv
```

in the in the `input` directory, and the files detailing the initial distribution,

```
x_PHAD_0.ssv
y_PHAD_0.ssv
z_PHAD_0.ssv
px_PHAD_0.ssv
py_PHAD_0.ssv
pz_PHAD_0.ssv
```

in the `output` directory. The files in the `output` directory have the same form as the files that will be written by each PHAD output step (determined by the input variable, `OUTPUT_RESOLUTION`). The files `phad-mpi.input` and `phad-mpi.input.relaunch` are also generated in the same folder as `make_beamlets_example.fox`.

To use this input data with PHAD, copy the `phad.fox` to the same directory where you ran `make_beamlets_example.fox`, and create empty subdirectory tmp (for temporary FMM files). Also, copy the fmm executable to the same directory. Execute the command

```
cosy phad.fox
```

to simulate the beamlet dynamics.

## 1.4 Visualization Utilities

The script `bmlt_mov.py` its auxiliary file `au.py` have been included in this example. The script takes the output data files from the PHAD run and creates density plots of the results like those shown in Figure 3. The basic syntax is

```
python bml_mov.py -s <start> -e <end> -i <increment>
```

where `-s` gives the output step to begin plotting (ie. x_PHAD_<s>.ssv), `-e` gives the last step to plot, and `-i` gives the increment between `s` and `e` that will be plotted. Uncommenting the `os.system` command at the bottom of the script will generate an mp4 movie of the plots using the general purpose visualization program ffmpeg.

| Variable Name | Description |
|---|---|
| `LOAD_BALANCE` | Load balance for FMM: 1 to turn on load balancing 0 to turn it off. |
| `q_value` | Maximum number of particles in a neighborhood allowed by FMM partitioning. |
| `FMM_ORDER` | FMM expansion order. |
| `FMM_EXECUTABLE` | Command to execute fmmcpp (including path). |
| `SIMO_ORDER` | Maximum allowed order for Simo integrator. |
| `NUM_OF_TIME_STEPS` | Number of PHAD timesteps to compute. |
| `OUTPUT_RESOLUTION` | Number of timesteps per output of positions and momentum. |
| `timestepsize` | PHAD timestep size ($\hat{t} = c\,t.$) |
| `SIMO_ACCURACY` | Accuracy of Simo integrator. |
| `DELTLIMIT` | Smallest timestep the Simo integrator may take ($\hat{t} = c\,t.$). |
| `BINS` | Number of particle bins in Simo integrator. |
| `BINNINGTYPE` | Type of binning 0 for equiparticle bins 1 for equi time width bins. |
| `max_num_of_boxes` | Estimate for the largest number of boxes or neighborhoods in the FMM partitioning. |
| `xmin` | Minimum transverse ($x$) position (for window). |
| `xmax` | Maximum transverse ($x$) position (for window). |
| `ymin` | Minimum transverse ($y$) position (for window). |
| `ymax` | Maximum transverse ($y$) position (for window). |

Table 1.2: User inputs for PHAD in `make_beamlets_example.fox`

# Chapter 2

# Beamlet Distribution Simulation

## 2.1 Boltzmann Velocity Selection

The Maxwell-Boltzmann distribution of particle speeds are given by

$$f(\mathbf{v}) = \left(\frac{m}{2\pi k_B T}\right)^{3/2} e^{-\frac{m||\mathbf{v}||^2}{2kT}},$$

where $T$ is temperature, $k$ is Boltzmann's constant, and $m$ is the particle's mass, and $||\mathbf{v}||^2$ is the square of the the velocity's Euclidean norm. Boltzmann's constant is

$$k_B = 1.380\,648\,52 \times 10^{-23}\,\mathrm{J\,K^{-1}}.$$

Notice that the maximum value of the distribution function is

$$\max(f) = \left(\frac{m}{2\pi kT}\right)^{3/2}.$$

The particle velocities can be generated using a 3D form of the Box-Muller transform (see Data File 2.1 for algorithm), which projects randomly sampled uniform data to a Gaussian distribution. A set of test values $\mathbf{u} = (u_1, u_2, u_3, u_4)$ is sampled from three uniform distributions taking values within $(0, 1)$ . The velocity vector $\mathbf{v}$ is then given by the following relation:

$$v_1 = \sqrt{-\frac{2k_B T \ln u_1}{m}} \cos(2\pi u_2)$$
$$v_2 = \sqrt{-\frac{2k_B T \ln u_1}{m}} \sin(2\pi u_2)$$
$$v_3 = \sqrt{-\frac{2k_B T \ln u_3}{m}} \cos(2\pi u_4)$$

A histogram of 8000 velocity samples generated via the Box-Muller algorithm 2.1 is shown in Figure 2.1 using 100 bins and compared to the analytic Boltzmann function.

```
FUNCTION BOLTZMANN MM KK TT TWICE_PI;
{Generates random numbers from 3D BOLTZMANN distribution
        using Box Muller transformation}
    VARIABLE U1 1;
    VARIABLE U2 1;
    VARIABLE U3 1;
    VARIABLE U4 1;

    VARIABLE Y1 1;
    VARIABLE Y2 1;
    VARIABLE Y3 1;

    VARIABLE CONST 1;


    IF TT=0;
        BOLTZMANN:=0&0&0;
    ELSEIF LO(1);
        RERAN U1;
        RERAN U2;
        RERAN U3;
        RERAN U4;
        U1:=abs(U1);
        U2:=abs(U2);
        U3:=abs(U3);
        U4:=abs(U4);

        CONST:=SQRT((KK*TT)/(MM));

        Y1:=CONST*SQRT(-2*LOG(U1))*COS(TWICE_PI*U2);
        Y2:=CONST*SQRT(-2*LOG(U1))*SIN(TWICE_PI*U2);
        Y3:=CONST*SQRT(-2*LOG(U3))*COS(TWICE_PI*U4);

        BOLTZMANN:=Y1&Y2&Y3;
    ENDIF;
ENDFUNCTION;
```

Data File 2.1.1: Box-Muller algorithm for generating a 3D Boltzmann distribution. Written for COSY INFINITY.

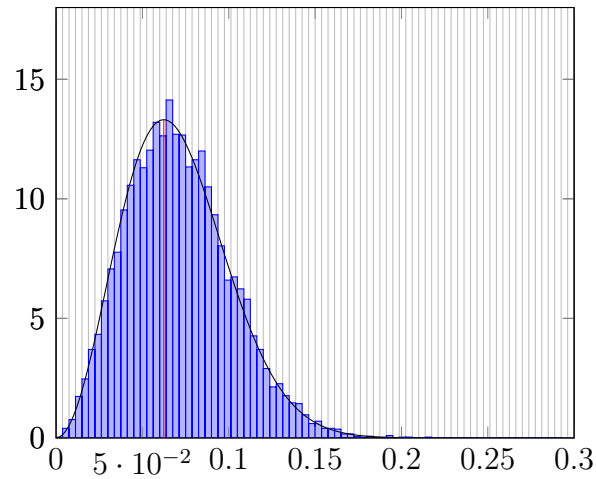


Figure 2.1: Boltzmann distribution velocity (black) compared to 8000 values sampled in 3D via the Box-Muller transformation. The distribution peak is shown by the red line.

## 2.2 Converting Velocity to Scaled Momentum

PHAD operates with certain scaled quantities for computational efficiency (see the PHAD Documentation). The values sampled from the Boltzmann velocity distribution must be converted to the PHAD scaled momentum $\hat{\mathbf{p}}_i$. The relativistic particle momentum is given by

$$\mathbf{p} = \gamma m \mathbf{v}, \tag{1}$$

where

$$\gamma = \frac{1}{\sqrt{1 - (||\mathbf{v}/c||^2)}}.$$

Multiplying numerator and denominator of the right-hand side of (1) by $c$ gives

$$\mathbf{p} = \frac{mc\mathbf{v}}{\sqrt{c^2 - ||\mathbf{v}||^2}}.$$

The scaled momentum of particle $i$, $\hat{\mathbf{p}}_i$ is $\mathbf{p}/(m_{\text{proton}}c)$ or

$$\hat{\mathbf{p}}_i = \frac{f_i \mathbf{v}}{\sqrt{c^2 - ||\mathbf{v}||^2}},$$

where $f_i = m_i/m_{\text{proton}}$. Thus $f_i = 1$ for protons, $f_i \approx 1/1800$ for electrons, and $f_i \approx A$ for heavier ions (where A is the ion's atomic mass).

# Chapter 3

# Ultracold Beamlet Dynamics

This example can be used to illustrate the power and scope of PHAD for accurately and efficiently capturing collisional particle dynamics, even with very dense beam currents. Running `make_beamlets_example.fox` with the given parameters will generate an array of nine electron beamlets (eight beamlets uniformly arrayed around a central beamlet). The electron dynamics (computed with PHAD) is shown in Figure 3, which illustrates both transverse spatial density and thermal domains. For more details on this application, see this IPAC 2021 presentation.

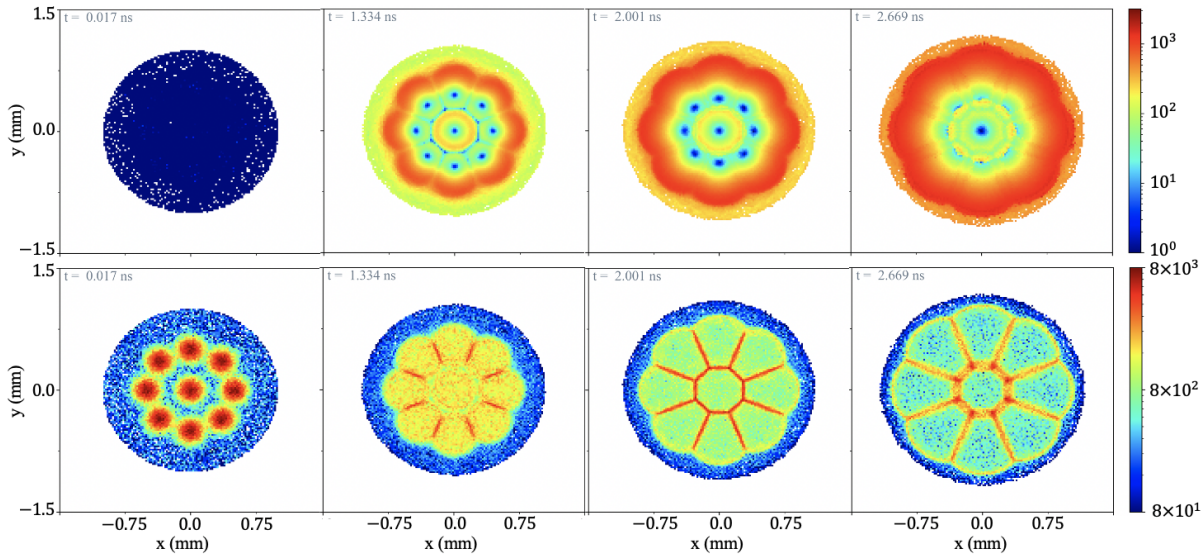


Figure 3.1: Electron temperature (K) above and charge density ($\mu$C/m$^3$) below at time $t = 0$ ns, 1.3 ns, 2.0 ns, 2.7 ns