GammaPBHPlotter: A public code for calculating the complete Hawking evaporation gamma-ray spectra from primordial black holes

John Carlini[1,*] and Ilias Cholis[1,†]

[1] *Department of Physics, Oakland University, Rochester, Michigan, 48309, USA*

(Dated: November 4, 2025)

Zenodo URL: https://zenodo.org/records/16944093

GitHub URL: https://github.com/jcarlini-dot/GammaPBHPlotter

PyPI URL: https://test.pypi.org/project/gammapbh

arXiv identifier: 1234.56789 [astro-ph.HE]

Abstract

We present `GammaPBHPlotter`, a public Python package for calculating and plotting the Hawking-radiation gamma-ray spectra of primordial black holes (PBHs) in the mass range of 5×10^{13}–10^{19} g. The code computes instantaneous and time-averaged spectra and includes all four photon components: direct Hawking emission, secondary emission from hadronization/decays (via `BlackHawk`+`PYTHIA`), final-state radiation, and in-flight annihilation. For speed, spectra at 56 logarithmically spaced PBH masses are precomputed and interpolated in log space. Users can generate spectra for monochromatic, Gaussian collapse, non-Gaussian collapse, lognormal, and custom-entered PBH mass distributions, visualize dN_γ/dE_γ and $E_\gamma^2 dN_\gamma/dE_\gamma$, and save results for comparison and downstream analysis.

CONTENTS

* jcarlini@oakland.edu, ORCID: orcid.org/0009-0002-2918-0300
† cholis@oakland.edu, ORCID: orcid.org/0000-0002-3805-6478

I. SUMMARY

Hawking radiation [1] remains an unobserved property of black holes. As the temperature of black holes is inversely proportional to the square of their mass, conventional stellar mass black holes are expected to emit too little radiation to ever be detected. However, primordial black holes (PBHs) that could have formed from the collapse of primordial perturbations in the early universe, can provide detectable signals [2]. PBHs with mass less than 5×10^{13} grams would have evaporated via Hawking radiation long before the present age of the universe. Upcoming gamma-ray telescopes such as e-ASTROGAM [3] and AMEGO-X [4] will be sensitive enough in the MeV range to detect the Hawking spectra of PBHs lying between this lower bound and 10^{19} grams. We have developed `GammaPBHPlotter`, an open-source software to simulate the exact gamma-ray spectra produced from different PBH mass-distributions. This document includes a complete overview of the underlying physics and operations of this software, as well as user instructions on how to best utilize it.

II. HAWKING SPECTRA

A. Modeling the emission components

The gamma-ray spectrum of a PBH within the relevant mass range consists of four primary components; direct/primary Hawking radiation, secondary radiation, final-state radiation, and in-flight annihilation.

Direct Hawking radiation accounts for all kinematically allowed elementary particles formed at the event Horizon [1], including gamma-ray photons. Secondary radiation originates from the decay of unstable particles and contributes significantly at lower energies. We rely on `BlackHawk` [5, 6] to evaluate the gamma-ray primary and secondary spectral components. `BlackHawk` uses `PYTHIA` [7] for the modeling of the hadronization and decay processes leading to the secondary spectra. Final-state radiation originates from relativistic electrons and positrons (e^+) and has a differential spectrum given by [8],

$$\frac{dN_\gamma^{\rm FSR}}{dE_\gamma} = \frac{\alpha}{2\pi}\int dE_{e^+}\frac{dN_{e^+}}{dE_{e^+}}\left(\frac{2}{E_\gamma}+\frac{E_\gamma}{E_{e^+}^2}-\frac{2}{E_{e^+}}\right)\left[\ln\left(\frac{2E_{e^+}+(E_{e^+}-E_\gamma)}{m_{e^+}^2}\right)-1\right], \tag{1}$$

where $\alpha \approx 1/137.037$ is the fine structure constant, E_{e+} is the kinetic energy of a given positron, E_γ is the energy of the emitted photon, $m_{e^+} = 0.511$ MeV is the rest mass of the electron, and $\frac{dN_{e+}}{dE_{e^+}}$ the differential spectrum of emitted electrons/positrons. In addition to the previously mentioned components, gamma-rays can be produced through pair-annihilation of positrons with interstellar medium electrons. This is known as in-flight annihilation and its differential spectrum is [8],

$$\begin{aligned}\frac{dN_\gamma^{\rm IA}}{dE_\gamma} &= \frac{\pi\alpha^2 n_H}{m_e}\int_{m_e}^{\infty} dE_{e^+}\frac{dN_{e^+}}{dE_{e^+}}\int_{E_{\rm min}}^{E_{e^+}}\frac{dE}{dE/dx}\frac{P_{E_{e^+}\to E}}{(E^2-m_e^2)} \\ &\times\left(-2-\frac{(E+m_e)(m_e^2(E+m_e)+E_\gamma^2(E+3m_e)-E_\gamma(E+m_e)(E+3m_e))}{E_\gamma^2(E-E_\gamma+m_e)^2}\right).\end{aligned} \tag{2}$$

We take $n_H = 1\,\mathrm{cm}^{-3}$ as the density of interstellar medium hydrogen (and by extension electrons). E_{e^+} is again the initial positron total energy, E is the final positron total energy, dE/dx is the rate of positron energy lost over distance via the Bethe-Bloch formula [9], E_γ is the resulting photon energy from annihilation, and $P_{E_{e^+}\to E}$ is the probability of a particular positron of a given initial and final energy to decay. This probability matrix can be calculated as [8],

$$P_{E_{e^+}\to E} = \exp\left(-n_H\int_E^{E_{e^+}}\sigma_{\rm ann}(E')\,\frac{dE'}{dx}\,dE'\right), \tag{3}$$

where σ_{ann} is the cross section of annihilation for positrons of a given energy.

In Fig. 1, we give the individual gamma-ray spectral components as well as their sum for a PBH of mass 3×10^{15} grams.

B. Interpolation

The process of performing these simulations can be quite cumbersome in terms of time and power constraints. In order to produce a more convenient experience for users, we opted to interpolate our data points rather than

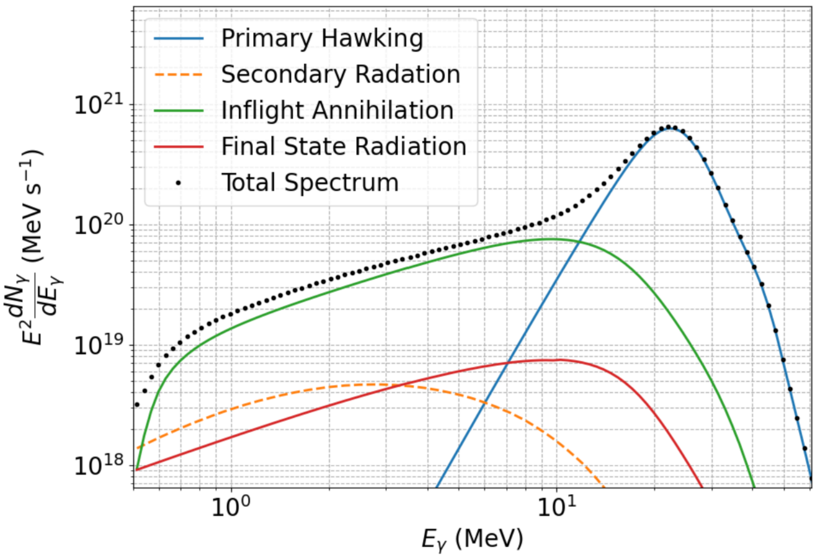


FIG. 1. The total gamma-ray spectrum of a 3×10^{15} grams PBH as well as its components.

render them in full for every PBH simulated. We precomputed the gamma-ray Hawking spectra of 56 different masses logarithmically spaced throughout our range. Those masses, in units of grams, are as follows.

$$5\times10^{13},\ 6\times10^{13},\ 7\times10^{13},\ 8\times10^{13},\ 9\times10^{13},$$
$$1\times10^{14},\ 1.5\times10^{14},\ 2\times10^{14},\ 3\times10^{14},\ 4\times10^{14},\ 5\times10^{14},\ 6\times10^{14},\ 7\times10^{14},\ 8\times10^{14},\ 9\times10^{14},$$
$$1\times10^{15},\ 1.5\times10^{15},\ 2\times10^{15},\ 3\times10^{15},\ 4\times10^{15},\ 5\times10^{15},\ 6\times10^{15},\ 7\times10^{15},\ 8\times10^{15},\ 9\times10^{15},$$
$$1\times10^{16},\ 1.5\times10^{16},\ 2\times10^{16},\ 3\times10^{16},\ 4\times10^{16},\ 5\times10^{16},\ 6\times10^{16},\ 7\times10^{16},\ 8\times10^{16},\ 9\times10^{16},$$
$$1\times10^{17},\ 1.5\times10^{17},\ 2\times10^{17},\ 3\times10^{17},\ 4\times10^{17},\ 5\times10^{17},\ 6\times10^{17},\ 7\times10^{17},\ 8\times10^{17},\ 9\times10^{17},$$
$$1\times10^{18},\ 1.5\times10^{18},\ 2\times10^{18},\ 3\times10^{18},\ 4\times10^{18},\ 5\times10^{18},\ 6\times10^{18},\ 7\times10^{18},\ 8\times10^{18},\ 9\times10^{18},\ 1\times10^{19}$$

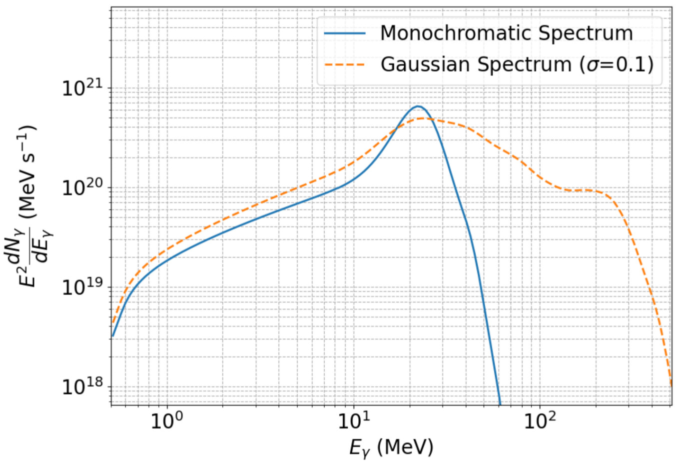
For each of those masses, we calculated the gamma-ray spectra of each of the four components (direct Hawking, secondary, inflight annihilation, final-state radiation) mentioned previously. With those data points, a linear spline in log space is used to interpolate the individual components and the total gamma-ray spectrum for any mass of PBH which lies within our range.

III. PBH DISTRIBUTION

Users can calculate the gamma-ray spectra from five types of PBH mass distributions. Those are, i) a monochromatic distribution with a mass to be set in the range of 5×10^{13} to 1×10^{19} grams, ii) a Gaussian distribution of PBH masses originating from a Gaussian distribution of density perturbations [10], iii) a more realistic non-Gaussian PBH mass distribution from [10], iv) a log-normal distribution of PBH masses, and v) a custom entered probability density function. In Fig. 2, we give the gamma-ray spectra from monochromatic and Gaussian PBH mass-distributions.

IV. SOFTWARE CONTENT

`GammaPBHPlotter` is a pure-Python package (tested with Python ≥ 3.9) that runs on Windows, Linux, and macOS. It exposes both a command-line entry point (`gammapbh`) and a module entry point (`python -m gammapbh`). All runtime dependencies are declared in the package metadata and are resolved automatically by `pip`. The code relies on: `colorama` [11], `NumPy` [12], `Matplotlib` [13], `tqdm` [14], and `SciPy` [15].



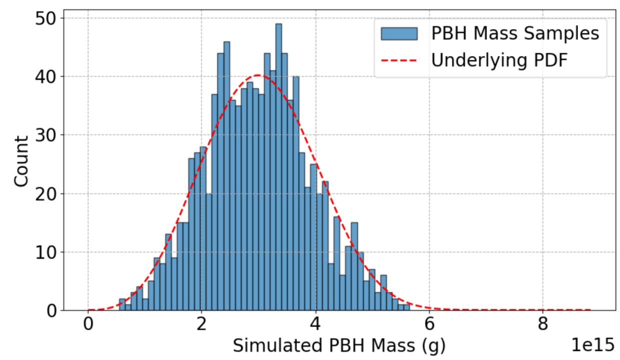


FIG. 2. Left: The total gamma-ray spectrum per PBH for a monochromatic PBH with mode mass 3×10^{15} g (blue) and for a Gaussian distribution with the same mode (orange). Right: Histogram of 1000 sampled PBH masses used to generate the figure on the left.

A. Package layout (installed tree)

```
src/gammapbh/
  | __init__.py          # version, public symbols
  | __main__.py          # enables ‘python -m gammapbh‘
  | cli.py               # CLI + core simulation/plotting routines
  | blackhawk_data/      # pre-rendered reference spectra (56 masses)
   results/             # auto-created output root for saved runs
```

The `blackhawk_data/` directory ships with the package and contains 56 reference mass folders (log-spaced over 5×10^{13}–10^{19} g); each folder includes the component spectra used for interpolation. The `results/` directory is created at runtime (if missing) and used as a writable root for all saved products (monochromatic and distributed runs).

B. Key modules and responsibilities

a. *`cli.py` (primary entry).*

- User interface: banner/start screen, main menu, input prompts, and back/quit navigation via a lightweight `BackRequested` exception and `user_input()` wrapper.
- Data discovery: `discover_mass_folders()` verifies the 56 mass subfolders and required files.
- Loading and alignment: `load_spectra_components()` ingests component tables, performs unit harmonization (GeV→MeV where applicable), and aligns secondaries/IFA/FSR to the primary energy grid.
- Interpolation: builds log–log splines with `RectBivariateSpline` (linear in $\log M$, cubic in $\log E$) for each component; protects against zeros via small floors.
- Physics helpers: `delta_l()`, `mass_function()` (Gaussian collapse), `mass_function_exact()` (non-Gaussian collapse), and `mass_function_lognormal()` (standard log-normal).
- Simulation flows:
  1. **Monochromatic** (`monochromatic_spectra()`): reuses exact pre-rendered bins when available; otherwise interpolates in $(\log M, \log E)$.
  2. **Distributed** (`distributed_spectrum(method)`): Gaussian, non-Gaussian, or log-normal sampling; mass draws are mapped to spectra (pre-rendered or interpolated) and averaged to yield $\langle dN/dE \rangle$.

3. **Custom PDF** (`custom_equation_pdf_tool()`): safe evaluation of user-provided $f(m)$ on a log-spaced mass grid, normalization, inverse-CDF sampling, and accumulation of the *total* spectrum.

- Stability guards: conservative tail handling for in-flight annihilation (flat-tail zeroing, large-drop detection) and an energy cutoff estimated from neighboring reference masses.
- Visualization: log–log panels for dN_γ/dE_γ and $E_\gamma^2 dN_\gamma/dE_\gamma$ with dynamic axes; distributed runs also show a 50-bin mass histogram with an analytic PDF overlay scaled to "expected counts per (log) bin."
- Persistence: standardized saving under `results/monochromatic`, `results/gaussian`, `results/non_gaussian`, `results/lognormal`, or `results/custom_equation`. Distributed runs write `distributed_spectrum.txt` and `mass_distribution.txt`; the custom-PDF flow additionally records the entered equation in `equation.txt`.

b. ***__main__.py.*** Thin launcher that delegates to `cli.main()`, enabling `python -m gammapbh` as an alternative to the `gammapbh` console script.

c. ***__init__.py.*** Exposes package version and minimal public surface for programmatic use.

C. Runtime I/O and reproducibility

All graphs are generated with `Matplotlib` using fixed energy limits (0.5 MeV–5 GeV) and y-limits scaled to each plot's peak for readability. Saved text files use plain ASCII with headers describing units and columns to facilitate downstream analysis and version control diffs. The combination of bundled reference spectra and deterministic interpolation ensures that identical inputs produce identical outputs across platforms (modulo BLAS-level floating-point variations).

V. INPUTS

Upon starting the software, the user is presented with a start screen. This includes the software's name, version, purposes, and other acknowledgments. Underneath the start screen is a main menu.

A. Menu Navigation

Whenever a user starts the program or completes a simulation, the user is presented with the following text. If the user enters 0, they will instantly close the program. If at any point, they wish to go back to a previous screen, they can enter `"b"` or "back" to do so.

$$\begin{aligned} Select :& 1 : Monochromatic\ spectra \\ & 2 : Distributed\ spectra\ (Gaussian\ Distribution) \\ & 3 : Distributed\ spectra\ (Non-Gaussian\ Collapse) \\ & 4 : Distributed\ spectra\ (Log-Normal\ Distribution) \\ & 5 : Distributed spectra (Custom\ mass\ PDF) \\ & 6 : View\ previous\ spectra \\ & 0 : Exit \end{aligned}$$

Entering the numbers 1-5, will direct the user to simulate the spectrum of PBHs of the listed distribution method.

B. User entered data

For monochromatic distribution, the user only needs to enter the peak masses to generate the desired spectra. One mass for a single simulation may be entered, but the user is also able to run multiple simulations at different peak masses if they are entered with comma separation. For example, if the entered text reads `"3.5e15, 4.2e14, 3e17"`, then the program would generate all 3 of those spectra in that order. For monochromatic spectra, this is the only value that the user needs to enter.

For any continuous distribution, the sample size (or N) also needs to be entered after the peak masses. N determines how many individual masses will be generated by the desired probability distribution function. A good minimum value to ensure precision is N ≥ 1000.

Finally, the software will ask for a particular value of the σ which denotes a standard-deviation parameter of the underlying variable. This variable is the unitless parameter of density perturbations (δ_l) for the Gaussian/non-Gaussian cases, and the natural log of PBH mass ($\ln M_{PBH}$) for the log-normal case). Due to specific limitations within the models developed by Biagetti et al.[10], the σ values used within the Gaussian and non-Gaussian collapse methods must be within a limited range.

With the Gaussian collapse method, σ_G must fall within this range,

$$0.03 < \sigma_G < 0.255.$$

With the non-Gaussian collapse method σ_G must fall within this range,

$$0.04 < \sigma_{NG} < 0.16.$$

For Log-normal distribution σ may be any value greater than 0. However, the binning structure for histograms begins to break down past usability at values greater than 2. Multiple different values of sigma may be entered with commas separating them similarly to mass.

Finally, the user is presented with the option to enter their own custom PDF. Any continuous function of mass (m) in grams with a positive value anywhere from 5×10^{13} to 1×10^{19} g may be used as the PDF to determine the mass spectra. Given the size of this range compared to the units, it is recommended to write the PDF of m as a function of $x = m/(10^{16}\,\text{gram})$ to prevent numerical issues on such a function. This function is compatible with `numpy` meaning expressions like `numpy.log(m)` or `numpy.pi` will be recognized even if entered without using the `"numpy"` or `"np"` prefix. The software detects any constants entered into the function and will request the user to define each of their values.

Once N PBH spectra are generated, these spectra are summed together before being divided by N to produce the average spectrum. This provides the user with the ability to observe the average contribution of individual PBHs and normalize it to the amplitude of spectra generated by a monochromatic distribution.

C. BlackHawk Data

Located within the folder labeled `blackhawk_data` there are 56 sub folders, each dedicated to one of the 56 reference masses and the pre-rendered spectral data associated with them. Each folder is named with the scheme (`"1.0e+14"`, `"1.5e+14"`, `"2.0e+14"`, etc.). Within each subfolder there are 9 .txt files which are listed and described in descending alphanumerical order.

`{mass}`: It lists the parameters used to input the data into BlackHawk and generate the direct and secondary radiation spectra.

`final_State_radiation_{prim/sec}`: Two files display the photon energy `E_gamma (MeV)` and the $\frac{dN_\gamma}{dE_\gamma}$ differential flux of the final state radiation `Final State Radiation in MeV`$^{-1}$ `s`$^{-1}$. The `final_State_radiation_prim` gives the spectrum calculated from the primary positrons, while the `final_State_radiation_sec` from the secondary positrons.

`inflight_annihilation_{prim/sec}`: Two files respectively give the inflight annihilation spectra calculated by primary and secondary positrons (`IFA_{primary/secondary}_masked) in MeV`$^{-1}$ `s`$^{-1}$.

`Instantaneous_primary_spectra`: This file is generated by `BlackHawk`. It lists in the header the first column as `energy/particle` which serves a similar, but more generalized, parameter to the `E_gamma` variable listed in previous files in units of GeV. From there, each subsequent column is labeled by the associated fundamental particle of which it represents the direct Hawking flux (photons, gluons, higgs, etc.) in units of GeV^{-1}s^{-1}. Note that as this file was generated by BlackHawk, it uses a different set of units from this software. The particle energy is multiplied by 10^3 to convert from GeV to MeV and particle flux is multiplied by 10^{-3} to convert GeV^{-1}s^{-1} to MeV^{-1}s^{-1}.

`Instantaneous_secondary_spectra`: This is a file generated to isolate the secondary flux of positrons and gamma-rays in units of MeV^{-1}s^{-1} compared to the particle's energy in units of MeV.

`Instantaneous_total_spectra`: This is the second file generated by `BlackHawk`. The only way it differs from the aforementioned `instantaneous_primary_spectra` file is that is that the spectra displayed here are the sum of the direct and secondary radiation. Note that as this file was generated by BlackHawk, it uses a different set of units from this software. The particle energy must be multiplied by 10^3 to convert from GeV to MeV and particle flux must be multiplied by 10^{-3} to convert GeV^{-1}s^{-1} to MeV^{-1}s^{-1} before either can be compared to any results from this software.

VI. GRAPHS

After all inputs have been entered, the user will see a second window open that displays their spectra in the form of a graph. The x-axis displays the energy of the emitted photons E_γ while the y-axis displays the Hawking radiation spectra $\frac{dN_\gamma}{dE_\gamma}$. The graphs will have limits of 500 keV to 5 GeV and the highest three orders of magnitude of the spectrum's peak along the y-axis.

Whenever a non-monochromatic spectrum is generated, a histogram will be displayed afterward. This shows the distribution of masses in 50 bins and the PDF that was used to generate them.

VII. SAVED FILES

Once all graphs have been displayed in a given simulation, the software will prompt the users asking if they wish to save any of the spectra. If they select `"y"` or `"yes"` in response, they will be presented with an indexed list of all spectra generated. The user needs only to enter the corresponding number or numbers (separated by commas) to decide which ones they desire to save. If `n` or `no` are entered, the user will return to the main menu.

Once saved, the user will read a confirmation message reading `"Saved → path"` with `"path"` referring to the file path to the destination folder. This folder will depend on which distribution method was used to generate the spectrum.

Monochromatic distribution goes to `.../results/monochromatic`
Gaussian distribution goes to `.../results/gaussian`
Log-normal distribution goes to `.../results/lognormal`
Non-Gaussian collapse goes to `.../results/non_gaussian`
Custom PDF distribution goes to `.../results/custom_equation`

Monochromatic spectra are each saved in one file under the name of `"mass_spectrum.txt"`. `"mass"` is a placeholder for the monochromatic mass value entered for that particular simulation. This file contains 7 columns. One to represent the photon energy in MeV, and 6 to represent the different components of the gamma-ray, Hawking radiation (Direct, secondary, inflight annihilation, final state radiation, and the total sum) in MeV^{-1} s^{-1}.

Saving Gaussian, non-Gaussian, or log-normal spectra instead produces two files inside a sub folder named `"mass_sigma.txt"`. `"sigma"` is another placeholder referring to the σ value entered for this particular simulation. Custom PDF spectra are instead placed inside `"mass_average.txt"` where `"average"` refers to the average mass of the given distribution of masses. For any distributed spectrum, The file named `"distributed_spectrum.txt"` has two columns comparing photon energy and the total gamma-ray spectrum. Meanwhile `"mass_distribution.txt"` has one column which lists every single mass generated in the simulation in units of grams. This second file also lists the distribution's mean mass in the header. In addition to these two files, the custom distribution spectrum also has a file called `"equation.txt"` which lists out the python expression entered to generate the spectrum.

VIII. ROUTINES

This section documents the interactive CLI routines provided by `GammaPBHPlotter`. All routines operate on the packaged BlackHawk tables under the internal `blackhawk_data/` directory and write outputs beneath the internal `results/` tree. Energies are in MeV, masses in grams, and spectra in MeV^{-1} s^{-1} unless stated otherwise. Plots are log–log by default.

A. Monochromatic Spectra

a. Purpose. Plot Hawking-radiation components and totals for one or more user-specified PBH masses M that lie within the pre-rendered grid domain $[\min M, \max M]$.

b. Workflow.

1. The tool discovers available mass folders to determine the admissible range $[\min M, \max M]$.

2. The user enters a comma-separated list of target masses M_i (no rounding or snapping is performed here).

3. For each M_i, component spectra are evaluated by bivariate splines in $(\log M, \log E)$: *Direct Hawking*, *Secondary*, *Inflight Annihilation (IFA)*, *Final State Radiation (FSR)*; a guarded trimming is applied to IFA high-energy tails (see §VIII F).

4. Per-mass figures are shown for components and totals, followed by an overlay of $E^2\,\mathrm{d}N/\mathrm{d}E$ across all requested masses (plus a summed curve).

5. On request, selected spectra are saved to `results/monochromatic/` with columns:

`E_gamma(MeV)`, `Direct`, `Secondary`, `Inflight`, `FinalState`, `Total`.

B. Distributed Spectra

The code supports three distribution families for PBH masses. In all cases the user provides (i) one or more *peak* masses, (ii) a target sample size N, and (iii) distribution-specific width parameters. Mass samples are generated, spectra are accumulated via $(\log M, \log E)$ splines (with IFA-tail guarding), and the following are displayed:

- An overlay of total $\mathrm{d}N/\mathrm{d}E$ for each parameter set.
- An overlay of $E^2\,\mathrm{d}N/\mathrm{d}E$ for each parameter set.
- For each set, a mass histogram (counts) with a *counts-scaled* analytic PDF overlay.

Saved outputs appear under method-specific subdirectories of `results/`, with:

- `distributed_spectrum.txt`: `E_gamma(MeV)`, `TotalSpectrum`.
- `mass_distribution.txt`: list of sampled masses (g).

1. Gaussian Collapse

a. Input parameters. A list of dispersions $\sigma_X \in [0.03,\, 0.255]$.

b. Model mapping. The routine samples over an internal, dimensionless grid $x \in [0.001,\, 1.30909]$. The collapse mapping uses

$$\delta_\ell(x) \;=\; \frac{8-\sqrt{\max\{\,0,\;64-96(\delta_c+y)\,\}}}{6}, \qquad y \equiv \left(\frac{x}{\kappa}\right)^{1/\gamma},$$

with default $\kappa = 3.3$, $\gamma = 0.36$, $\delta_c = 0.59$. A proxy mass function in δ_ℓ-space is evaluated (up to an overall constant) and truncated to positive, finite values:

$$\phi_{\mathrm{G}}(\delta_\ell) \;\propto\; \frac{1}{\sqrt{2\pi}\,\sigma_X}\,\exp\!\left(-\frac{\delta_\ell^2}{2\sigma_X^2}\right)\,\frac{\delta_\ell-\frac{3}{8}\delta_\ell^2-\delta_c}{\gamma\,|1-\frac{3}{4}\delta_\ell|}\,.$$

The resulting discrete PDF over x is normalized; its mode x_{mode} is found and the grid is scaled so that the requested peak mass M_{peak} is realized via $M = x\,M_{\mathrm{peak}}/x_{\mathrm{mode}}$.

2. Non-Gaussian Collapse

a. Input parameters. A list of $\sigma_X \in [0.04,\, 0.16]$ with a fixed ratio $\sigma_Y/\sigma_X = 0.75$.

b. Model mapping. Using the same $\delta_\ell(x)$ map and constants $(\kappa, \gamma, \delta_c)$ as above, the routine evaluates a Biagetti–et al.–like non-Gaussian proxy (up to constants):

$$\phi_{\mathrm{NG}}(\delta_\ell) \;\propto\; \frac{e^{-1/(2\sigma_Y^2)}\,\sigma_X}{2\pi\,A^{3/2}}\left[2\sigma_Y\sqrt{A}+\sqrt{2\pi}\,\sigma_X\,\exp\!\left(\frac{\sigma_X^2}{2\sigma_Y^2(\sigma_X^2+2(\sigma_Y\delta_\ell)^2)}\right)\mathrm{erf}\!\left(\frac{\sqrt{2}\,\sigma_X}{\sqrt{A}}\right)\right]\mathcal{J}(\delta_\ell),$$

where $A = \sigma_X^2 + (\sigma_Y\delta_\ell)^2$ and the mapping Jacobian $\mathcal{J}(\delta_\ell) = \dfrac{\delta_\ell-\frac{3}{8}\delta_\ell^2-\delta_c}{\gamma\,|1-\frac{3}{4}\delta_\ell|}$. As for the Gaussian case, the discrete PDF over x is normalized, mode-matched, and scaled to produce the requested M_{peak}.

3. Log-normal Distribution

a. Input parameters. A list of log-space standard deviations $\sigma > 0$. For a given $M_{\rm peak}$, the code chooses

$$\mu_{\rm eff} \;=\; \ln M_{\rm peak} + \sigma^2,$$

so that the *mode* of the log-normal equals $M_{\rm peak}$. Masses are drawn via LogNormal$(\mu_{\rm eff}, \sigma)$. The mass-PDF used for overlays is

$$p_{\rm LN}(M; \mu_{\rm eff}, \sigma) \;=\; \frac{1}{M\,\sigma\,\sqrt{2\pi}}\,\exp\!\left[-\frac{(\ln M - \mu_{\rm eff})^2}{2\sigma^2}\right].$$

C. Custom Equation Mass-PDF Tool

a. Purpose. Construct a mass PDF from a user-provided right-hand-side expression $f(m)$ (with m in grams), normalize it on the discovered domain $[\min M, \max M]$, sample N masses, and produce the *average total* spectrum.

b. Expression handling.

- The input may include a prefixed `f(m)=`, which is stripped automatically.
- A restricted, NumPy-like safe namespace is used ($\log, \exp, \sqrt{}, \sin, \cos, \tan, \arctan, \mathrm{erf}, \mathrm{abs}, \mathrm{clip}, \min/\max, \pi, e$).
- Unknown identifiers are detected; the user is prompted to supply their numeric values.

c. Numerical steps.

1. Build a log-spaced grid of m (length 2×10^4) over $[\min M, \max M]$, evaluate and clip $f(m) \geq 0$, then normalize $p(m) = f(m)/\int f$.
2. Construct the CDF by trapezoidal integration and sample N masses by inverse-CDF.
3. For each mass, evaluate components via splines (with IFA-tail guarding) and accumulate the *total* spectrum; components are not stored separately.
4. Plot $\mathrm{d}N/\mathrm{d}E$, $E^2\,\mathrm{d}N/\mathrm{d}E$, and a log-binned mass histogram with a smooth *counts-scaled* PDF line $\propto N\,p(m)\,m\,\mathrm{d}\ln m$.

d. Saved outputs. In `results/custom_equation/{median_mass}_custom_eq_*/`:

- `equation.txt` (commented variable assignments followed by the expression).
- `samples_sorted.txt` (masses in grams).
- `distributed_spectrum.txt` with `E_gamma(MeV)`, `TotalSpectrum`.

D. Viewer: Previously Saved Spectra (Queue-Based)

a. Overview. The viewer builds a *queue* of items to plot. Selecting items only adds them to the queue; pressing `0` plots:

1. Spectral overlays: $\mathrm{d}N/\mathrm{d}E$ and $E^2\,\mathrm{d}N/\mathrm{d}E$.
2. Histograms for distributed/custom runs (monochromatic items have no histograms).

After plotting, the queue is cleared.

b. Monochromatic in the Viewer.

- *No listing of prior monochromatic files.* The user enters masses directly within $[\min M, \max M]$.
- Spectra are obtained *by interpolation in* $(\log M, \log E)$ *without snapping or rounding*, then queued for overlay plots.

c. Distributed/Custom in the Viewer.

- Saved run folders are listed under their method-specific results roots.
- Selecting indices queues their saved total spectra for overlays and records the run info for histogram reconstruction:
  - Gaussian/Non-Gaussian/Log-normal: histograms are rebuilt from `mass_distribution.txt` with analytic, counts-scaled PDF overlays.
  - Custom equation: histograms are rebuilt from `samples_sorted.txt` and, when available, the analytic PDF reconstructed from `equation.txt`.
- Histogram figures for distributed/custom runs are rendered at larger size (approximately 10×6).

E. File Layout and Naming

- **Monochromatic:** `results/monochromatic/`$\rightarrow$ `{mass}_spectrum.txt` with:

  `E_gamma(MeV)`, `Direct`, `Secondary`, `Inflight`, `FinalState`, `Total`.

- **Distributed:** `results/{gaussian|non_gaussian|lognormal}/{tag}/`
  - `distributed_spectrum.txt`: `E_gamma(MeV)`, `TotalSpectrum`.
  - `mass_distribution.txt`: sampled masses (g).
- **Custom equation:** `results/custom_equation/{median_mass}_custom_eq_*/`
  - `equation.txt`, `samples_sorted.txt`, `distributed_spectrum.txt`.

F. Numerical and Plotting Conventions

- **Interpolation.** All spectral interpolation is performed by `RectBivariateSpline`, linear in $\log M$ and cubic in $\log E$. A small floor ($\sim 10^{-300}$) avoids $\log(0)$, then values below 10^{-299} are reset to zero for plotting/saving.
- **IFA-tail guarding.** The right-edge of IFA is stabilized by zeroing terminal flat segments, clipping ultra-steep downward jumps in $\log_{10}$ space, and enforcing $S_{\rm IFA}(E \geq E_{\rm cut}) = 0$, with $E_{\rm cut}$ chosen from pre-rendered IFA energy maxima bracketing the target mass.
- **Overlays.** For each set/mass, both $\mathrm{d}N/\mathrm{d}E$ and $E^2\,\mathrm{d}N/\mathrm{d}E$ overlays are shown; axis limits are set by per-figure peak heuristics.
- **Histogram sizing.** Mass histograms in the viewer for distributed/custom runs are rendered at larger size (about 10×6) to match generation-time figures.

G. Parameter Bounds and Defaults

- Gaussian collapse: $\sigma_X \in [0.03, 0.255]$.
- Non-Gaussian collapse: $\sigma_X \in [0.04, 0.16]$ with fixed $\sigma_Y/\sigma_X = 0.75$.
- Log-normal: $\sigma > 0$ (log-space standard deviation), with $\mu_{\rm eff} = \ln M_{\rm peak} + \sigma^2$.
- Collapse-mapping constants: $\kappa = 3.3$, $\gamma = 0.36$, $\delta_c = 0.59$.

IX. CONCLUSION

GammaPBHPlotter is an open source program designed first and foremost to allow for an intuitive, comprehensive, and efficient means of calculating the gamma-ray Hawking spectra of low-mass primordial black holes in the range of 5×10^{13} to 10^{19} grams. It does this by interpolating pre-rendered spectral data of PBHs as a product of their mass. Then, via a number of distribution methods, users can calculate the expected gamma-ray spectrum PBHs would emit under different circumstances. With these pieces of data at hand, researchers can now more easily compare observed excess gamma radiation to probe for potential PBH signals.

X. ACKNOWLEDGMENTS

We acknowledge the use of BlackHawk [5, 6] and suggest to users of this software to cite BlackHawk together with our work. This material is based on work supported by the U.S. Department of Energy, Office of Science, Office of High Energy Physics, under Award No. DE-SC0022352.

[1] S. W. Hawking, Nature (London) **248**, 30 (1974).
[2] B. Carr, F. Kuhnel, and M. Sandstad, Phys. Rev. D **94**, 083504 (2016), arXiv:1607.06077 [astro-ph.CO].
[3] M. Tavani *et al.* (e-ASTROGAM), JHEAp **19**, 1 (2018), arXiv:1711.01265 [astro-ph.HE].
[4] R. Caputo *et al.*, J. Astron. Telesc. Instrum. Syst. **8**, 044003 (2022), arXiv:2208.04990 [astro-ph.IM].
[5] A. Arbey and J. Auffinger, Eur. Phys. J. C **81**, 910 (2021), arXiv:2108.02737 [gr-qc].
[6] BlackHawk Project, "Blackhawk," https://blackhawk.hepforge.org (2025), accessed 26 Aug 2025.
[7] T. Sjöstrand, S. Ask, J. R. Christiansen, R. Corke, N. Desai, P. Ilten, S. Mrenna, S. Prestel, C. O. Rasmussen, and P. Z. Skands, Comput. Phys. Commun. **191**, 159 (2015), arXiv:1410.3012 [hep-ph].
[8] C. Keith, D. Hooper, T. Linden, and R. Liu, Phys. Rev. D **106**, 043003 (2022), arXiv:2204.05337 [astro-ph.HE].
[9] H. A. Bethe and J. Ashkin, in *Experimental Nuclear Physics*, edited by E. Segrè (John Wiley & Sons, New York, 1953) p. 253.
[10] M. Biagetti, V. De Luca, G. Franciolini, A. Kehagias, and A. Riotto, Phys. Lett. B **820**, 136602 (2021), arXiv:2105.07810 [astro-ph.CO].
[11] J. Hartley, "Colorama," https://pypi.org/project/colorama/ (2022), python package, version 0.4.6.
[12] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, Nature **585**, 357 (2020).
[13] J. D. Hunter, Computing in Science & Engineering **9**, 90 (2007).
[14] C. O. da Costa-Luis and tqdm developers, "tqdm: A fast, extensible progress bar for python and cli," https://doi.org/10.5281/zenodo.11107065 (2024), software release (v4.66.4).
[15] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and S. . Contributors, Nature Methods **17**, 261 (2020).