1 Problem set 1: Classic radiation transport

1.1 Classic formal solution of the radiative transfer equation

The classical formal solution of the plane parallel radiative transfer equation is given by

$$\begin{aligned} J(\tau) &= \frac{1}{2}\left\{\int_\tau^\infty E_1(t-\tau)S(t)\,dt + \int_0^\tau E_1(\tau-t)S(t)\,dt\right\} \\ &= \frac{1}{2}\int_0^\infty E_1(|t-\tau|)S(t)\,dt \end{aligned}$$

with the standard form form of the 1^{st} exponential integral:

$$E_n(x) = \int_1^\infty t^{-n}\exp(-xt)\,dt$$

Write a python class (will be useful later) to

1. Generate a logarithmic τ grid: $\log_{10}$ spaced points from τ_{min} to τ_{max} for N_{layer} total points (all parameters on instantiation), set the outermost point $\tau = 0$.
2. Prepare class variables (numpy arrays) for B, J, S, H, K, etc. Set $B = J = S = 1$ on instantiation (we'll start testing with something simple).
3. Write a method to realize the kernel for the classic formal solution, i.e., $E_1(|t-\tau|)S(t)$, for any argument t (note: the modules sympy and in particular mpmath may be useful to save yourself work).
4. Write a method to compute $J(\tau)$ for all points of your optical depth grid for a given, fixed, $S(\tau)$. Note: again sympy and mpmath may be useful for easy numerical quadrature but beware.
5. plot the resulting $J(\tau)$

1.2 Classic Λ iterations

Use your code to implement a Λ iteration and calculate the results for $S = (1-\epsilon)J + \epsilon B$ and $\epsilon = 1,\ 0.5,\ 0.1,\ 10^{-2},\ 10^{-4}$ (independent of τ for simplicity). Plot the convergence behaviors (corrections $\max(|\Delta J|/J)$ as functions of iteration number). Limit the number of iterations to 100 ... and profile the runs to see where the time is burnt (module 'cProfile' works well enough).