1 Implementation

Define

$$A[g(\omega)] = \int_{E_0}^{\infty} dE\, e^{\alpha E} \left| \sum_{t=0}^{t_{\max}} g_t(\omega) b_T(t+1,E) - \Delta_\sigma(E,\omega) \right|^2 , \tag{1}$$

where

$$b_T(t,E) = e^{-(T-t)E} + e^{-tE} . \tag{2}$$

Eq. (2) is defined as `gte` in `core.py`. We also define

$$B[g(\omega)] = \frac{1}{B_{\text{norm}}} \vec{g}(\omega)^T \cdot \text{Cov}_d \cdot \vec{g}(\omega) . \tag{3}$$

The smearing kernel will be

$$\Delta_\sigma(E,\omega) = \frac{e^{\frac{-(E-\omega)^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma} \tag{4}$$

Other definitions used in the code follows, all found in `core.py`. These are:

`A0_mp` and `A0E_mp`

$$A_0(\omega) \equiv A[0](\omega) = \int_{E_0}^{\infty} dE\, e^{\alpha E} \Delta_\sigma(E,\omega)^2 = \frac{e^{\frac{\alpha^2\sigma^2}{4}+\alpha\omega} \left(\text{erf}\left(\frac{\alpha\sigma^2+2\omega-2e_0}{2\sigma} \right) + 1 \right)}{4\sqrt{\pi}\sigma} \tag{5}$$

`ft_mp`

$$\begin{aligned} f_t(\omega) &= \int_{E_0}^{\infty} dE\ \Delta_\sigma(E,\omega)\, b_T(t,E)\, e^{\alpha E} \qquad (6) \\ &= \frac{1}{2} \Big\{ \ e^{\frac{1}{2}(\alpha+t-T)\left(\sigma^2(\alpha+t-T)+2\omega\right)} \left(\text{erf}\left(\frac{\sigma^2(\alpha+t-T)+\omega-e_0}{\sqrt{2}\sigma} \right) + 1 \right) \\ &+ e^{\frac{1}{2}(\alpha-t)\left(\sigma^2(\alpha-t)+2\omega\right)} \text{erfc}\left(\frac{\sigma^2(t-\alpha)-\omega+e_0}{\sqrt{2}\sigma} \right) \Big\} \end{aligned}$$

In the code, we express $f_t(\omega)$ by means of the following function called `generalised_ft`:

$$\tilde{f}_t(\omega) = e^{\frac{1}{2}(\alpha-t)\left(\sigma^2(\alpha-t)+2\omega\right)} \text{erfc}\left(\frac{\sigma^2(t-\alpha)-\omega+e_0}{\sqrt{2}\sigma} \right) , \tag{7}$$

so that we can write

$$f_t(\omega) = \frac{\tilde{f}_t(\omega) + \tilde{f}_{T-t}(\omega)}{2} . \tag{8}$$

`Smatrix_mp`

$$\mathtt{S}_{tr} = \frac{e^{E_0(\alpha-r-t-2)}}{t+r+2-\alpha} + \frac{e^{E_0(\alpha+r+t+2-2T)}}{2T-t-r-2-\alpha} + \frac{e^{E_0(\alpha+r-t-T)}}{T+t-r-\alpha} + \frac{e^{E_0(\alpha-r+t-T)}}{T-t+r-\alpha} \tag{9}$$

We also have

$$\mathtt{B}_{tr} = \text{Cov}_{tr} . \tag{10}$$

$B_{\mathrm{norm}} = C(1)$ can be used to make $B[g]$ dimensionless. The minimisation then amounts to solve the following linear system

$$\vec{g} = \left(\mathtt{S} + \frac{\lambda\, A_0(\omega)}{(1-\lambda) B_{\mathrm{norm}}(\omega)}\, \mathtt{B} \right)^{-1} \vec{\mathtt{f}}\,. \tag{11}$$

In practice, W is rescaled

$$W[g] = (1-\lambda) \left\{ \frac{A[g]}{A_0} + \frac{\lambda}{(1-\lambda)} \frac{B[g]}{B_{\mathrm{norm}}} \right\}, \tag{12}$$

and we redefine $\lambda' \in (0, \infty)$.

1.1 Input parameters

- `datapath`: path to data file.
- `outdir`: location where output is saved.
- `prec`: numerical precision for the arithmetic, approximatively given in the number of digits.
- `tmax`: number of correlators that will be used for the reconstruction. By default, this is $T-1$ for open and $T/2$ for periodic boundaries, T being the time extent of the lattice. In this way, one uses $c(t)$ with $t \in [1, T-1]$ for open and $[1, T/2]$ for periodic boundaries. $t = 0$ is always excluded.
- `sigma`: smearing radius of the Gaussian kernel.
- `emin`, `emax`, `ne`: The reconstruction is performed at `ne` points in energy, spacing from `emin` to `emax`. The points are equally spaced. `emin` and `emax` are given in unit of `mpi`.
- `nboot`: number of bootstrap samples
- `e0`: value of E_0 used in the equations above.
- `mpi`: a unit of mass, e.g. m_π to be used in plots.