Sigmond Analysis

Andrew Hanlon

February 18, 2021

1 Introduction

This document describes the use of the `run-sigmond` program for running an analysis using the `sigmond` program. This program requires input from the user in order to know what to run, which is achieved by providing a configuration file to the program.

2 Command-Line Options

Given a configuration file, the `run-sigmond` program can be executed via

```
run_sigmond.py -c CONFIG-FILE
```

Use the `-h` or `--help` flag to see other possible options. For instance, to increase output you can include the `--verbose` flag, or, to increase output further, the `--debug` flag.

2.1 Installation

2.2 Overview of the input files

The format for the configuration files input into the `run-sigmond` program follows that of a YAML file.[1] Reading/Writing these configuration files is done via the `PyYAML` Python library.[2] The files consist of blocks, of which six are recognized by `run-sigmond`: `Initialize`, `Execute`, `MCBinsInfo`, `MCSamplingInfo`, `MCObservables`, and `Tasks` (not all blocks are required, see sections below for more details). Other, user defined, blocks are also allowed for the purpose of referring to them later for brevity. The form of these blocks will now be discussed.

2.3 Initialize Block

This block is required, and has the form

```
Initialize:
  project_directory: /home/user/analysis/my-project/
  ensembles_file: /home/user/analysis/ensembles.xml   # optional
  echo_xml: true    # optional, default is false
  raw_data_directories: # optional
    - /home/user/data/my-correlators1/
    - /home/user/data/my-correlators2/
```

[1] `http://yaml.org`
[2] `https://pyyaml.org/wiki/PyYAMLDocumentation`

The project_directory key is required and specifies where all files for this project will be created. Many tasks will rely on this as the base directory, so make sure that separate runs within a given project have the same project directory. Currently, the user has no say in how the files will be organized within the project directory. The key ensembles_file is optional and specifies the file that sigmond should use for determining ensemble information. If this key is missing, then the default file specified during compilation will be used by sigmond. If the key echo_xml is set to true, then sigmond will output the input XML file used in each log file.

If any raw data (*i.e.* data not created by sigmond) is required for the executition of the tasks specified, then these data files can be automatically discovered based on the directories passed to the raw_data_directories key and the channels specified in the tasks. One word of caution: when `run-sigmond` searches for data, correlator infos (for correlator data) or operator infos (for VEV data) are used as keys in a file map. Therefore, if the same correlator info occurrs in different files, then the first correlator info discovered is the only one saved in the map. However, you do have some control over this. For instance, in the example above, if the same correlator info exists in `my-correlators1` and `my-correlators2`, then the correlator in `my-correlators1` will be saved because it came first. However, this only works for basic LapH files, because the constructors for `BinsGetHandler` and `SamplingsGetHandler` take sets of files, which don't preserve order (maybe I want a list of these handlers; one for each file?). Alternatively, one can specify the exact data files to use within the MCObservables block, and these will always take precedence over data discovered via the raw_data_directories key.

One final important note: searching for files is based on file extension. Basic LapH files must have a non-negative integer for an extension, bin files must have a `bin` extension, and sampling files must have a `smp` extension. Be careful to make sure there are no non-sigmond readable files in the raw data directories, because these could lead to problems being mistaken for sigmond readable files.

2.4 Execute Block

The Execute block is optional. If it is missing, then the `sigmond` input XML files will be produced and nothing else will be done. This block specifies how sigmond should be run. That is, locally

```
Execute:
  mode: local
  sigmond_batch: /home/user/sigmond/build/sigmond_batch
  max_simultaneous: 4 # optional, default is number of processors on machine
```

where you must specify the location of the `sigmond` executable with the sigmond_batch key. You can also specify the maximum number of simulaneously executed jobs with the max_simultaneous key (the default is 1). Or, you can run your jobs on a PBS cluster

```
Execute:
  mode: PBS
  sigmond_batch: /home/user/sigmond/build/sigmond_batch
  email: user@mail.com
  queue: green
  walltime: 1:30:00 # optional
  cputime: 2:00:00 # optional, default is WallTime
  nodes: 4 # optional, default is 1
  processes_per_node: 16 # optional, default is 1
  memory: 850mb # optional
  extra: |
    cd /some/directory/
    export VAR=10
```

If the walltime and/or memory keys are not provided, then they will be determined automatically based on the type of tasks to be performed. The commands specified with the extra key will be placed in the PBS runscript before the execution of sigmond. `run-sigmond` will run one sigmond job per core.

Therefore, if you specify 4 nodes and 16 processes per node, then 4×16 sigmond processes will be placed in a single runscript. More runscripts will be created if more `sigmond` input files exist.

2.5 MCBinsInfo Block

The `MCBinsInfo` block is required and has the following form

```
MCBinsInfo:
  ensemble_id: phirho_s14_t48_mp0150_mr0400_lm0050_0300
  num_measurements: 12800 # optional
  num_streams: 1 # optional
  n_x: 14 # optional
  n_y: 14 # optional
  n_z: 14 # optional
  n_t: 48 # optional
  rebin: 4 # optional, default is 1
  omissions: [122, 343, 781, 1001] # optional, default is empty list
```

If the `ensemble_id` is known by `run-sigmond`, then that is the only required key, otherwise all other keys are required. The current list of known EnsembleIDs is: clover_s24_t128_ud840_s743, clover_s24_t128_ud860_s743, clover_s32_t128_ud860_s743, clover_s16_t128_ud840_s743, U103, H101, and B450.

2.6 MCSamplingInfo Block

The MCSamplingInfo block specifies the resampling mode to use and has the following form

```
MCSamplingInfo:
  mode: Bootstrap # or Jackknife
  number_resampling: 825 # optional, default is 1000
  seed: 1050 # optional, default is 0
  boot_skip: 250 # optional, default is 0
  precompute: true # optional, default is false
```

This block is optional, and if it is missing then Jackknife mode is used. Further, if the mode is specified to be Jackknife, then all other keys in the block are ignored.

2.7 MCObservables Block

This block is optional. All data files can be determined automatically by `run-sigmond`: raw data files can be found automatically by using the `RawDataDirectory` key in the `Initialize` block, and all data files produced by `sigmond` can be found automatically if the `ProjectDirectory` key in the `Initialize` block is consistent between runs. However, you may also choose to add any data files manually using this block. There is one exception: reweighting files must be specified here. They cannot be found automatically. Further, you can use data specifications in this block. The form of this block is as follows

```
MCObservables:
  BLCorrelatorData:
    - FileNameStub: /path/to/data/corr_A1_P001
      MinFileNumber: 0
      MaxFileNumber: 123
      Overwrite: true # optional, default is false
    - FileNameStub: /path/to/data/corr_E_P010
      MinFileNumber: 10
      MaxFileNumber: 99
      Overwrite: true # optional, default is false
  BLVEVData:
    - FileNameStub: /path/to/data/vev_A1g_P000
      MinFileNumber: 1
      MaxFileNumber: 8
      Overwrite: true # optional, default is false
  BinData:
```

```
    - /path/to/bin/data/some_bins.bin
  SamplingData:
    - /path/to/sampling/data/some_sampling_1.smp
    - /path/to/sampling/data/some_sampling_2.smp
    - /path/to/sampling/data/some_sampling_3.smp
  ReweightingData:
    Format: OPENQCD # or OPENQCD_12, or ASCII
    Files:
      - /path/to/reweighting_factors/reweighting_factors_1.dat
      - /path/to/reweighting_factors/reweighting_factors_2.dat
  UseCheckSums: true # default is false
  Specifications: # optional, example shown here
    - Correlator:
        Source: kaon P=(1,0,0) A1_1 SS_0 # a BLOpString
        Sink: kaon P=(1,0,0) A1_1 SS_0 # a BLOpString
    - Correlator:
        Source: kaon P=(1,0,0) A1_1 SS_2 # a BLOpString
        Sink: kaon P=(1,0,0) A1_1 SS_2 # a BLOpString
    - VEV: eta P=(0,0,0) A1g SS_0 # a BLOpString
```

The valid keys inside the `Specifications` key are: `Correlator`, `CorrelatorWithVEV`, `VEV`, `CorrelationMatrix`, `CorrelationMatrixWithVEVs`, `HermitianCorrelationMatrix`, `HermitianCorrelationMatrixWithVEVs`, `ObsBins`, and `ObsSamplings`.

2.8 Tasks

This section will describe the different ways to specify tasks using `run-sigmond`. The `Tasks` takes a set of keys specifying `run-sigmond` tasks.[3] The possible keys corresponding to `run-sigmond` tasks are: `ConvertData`, `CheckData`, `AverageCorrelators`, `ViewData`, `Prune`, `Rotate`, `CorrelatorFits`, `TwoCorrelatorFits`, `RatioFits`, `AnisotropyFit`, and `Spectrum`. Each of these will be described in the sections below, but first we go discuss how to specify `sigmond` tasks.

2.8.1 Check Data

This task is used to perform checks on the raw data.

```
CheckData:
  channels:
    - isospin: doublet
      strangeness: 1
      momentum: [0,0,0]
      irrep: A1g
      irreprow: 1
    - isospin: doublet
      strangeness: 1
      momentum: [0,0,1]
      irrep: E
      irreprow: 2
  outlier_scale: 12 # optional, default is 10
  hermitian: false # optional, default is true
  subtractvev: false # optional, default is true
```

The `outlier_scale` is used in the check for outliers in the data (see `sigmond` documentation for more details).

[3] Note that there is a distinction between `run-sigmond` tasks and `sigmond` tasks. `sigmond` tasks refer to individual tasks that `sigmond` runs. `run-sigmond` tasks are larger tasks that involve multiple `sigmond` tasks and possibly multiple `sigmond` input XML files. This distinction is important to keep in mind while reading this documenation.

2.8.2 Average Correlators

This task is used to average data over irrep rows and/or channels with equivalent momentum. The task takes a list of channels. The members of the list are NOT being averaged together. Each member corresponds to the resulting averaged channel. For each averaged channel, `run-sigmond` uses all data it can find to average over and create that channel. For each channel, you can also specify a list of operators and a coefficient to use in the averaging. All operators not present are assumed to have a coefficient equal to 1.0. An example for the form of this task is

```
AverageCorrelators:
  averaged_channels:
    - isospin: doublet
      strangeness: 1
      momentum_squared: 1
      irrep: A1
      irreprow: 1
      coefficients:
        - operator: kaon P=(0,0,1) A1_1 SS_0
          coefficient: -1.0
        - operator: kaon P=(0,0,1) A1_1 SS_1
          coefficient: -2.0
        - operator: kaon P=(0,0,1) A1_1 SS_2
          coefficient: 2.2
    - isospin: doublet
      strangeness: 1
      momentum: [0,0,0]
      irrep: T1u
      coefficients:
        - operator: kaon P=(0,0,1) E_1 SS_0
          coefficient: -1.0
        - operator: kaon P=(0,0,1) E_1 SS_1
          coefficient: -2.0
        - operator: kaon P=(0,0,1) E_1 SS_2
          coefficient: 2.2
    - isospin: doublet
      strangeness: 1
      momentum_squared: 3
      irrep: E
      coefficients:
        - operator: kaon P=(0,0,1) E_1 SS_0
          coefficient: -1.0
        - operator: kaon P=(0,0,1) E_1 SS_1
          coefficient: -2.0
        - operator: kaon P=(0,0,1) E_1 SS_2
          coefficient: 2.2
```

Notice how the specification of the channels is what determines the type of averaging to be done. If the `momentum_squared` key is present, then an average over equivalent momentum channels is done (in this case the `momentum` key cannot be present). If the `irreprow` key is missing, then an average over irrep rows is performed. Thus, in the example above, the first channel averages over $P^2 = 1$ frames, the second channel averages over irrep rows, and the third channel averages over both irrep row and equivalent momentum.

2.8.3 View Data

This task will produce PDF files showing the correlators and effective energies for the channels specified. The required block for this task is of the form

```
ViewData:
  channels:
    - isospin: doublet
      strangeness: 1
      momentum: [0,0,0]
      irrep: A1g
```

```
      irreprow: 1
    - isospin: doublet
      strangeness: 1
      momentum_squared: 1
      irrep: A1
      irreprow: 1
    - isospin: doublet
      strangeness: 1
      momentum_squared: 3
      irrep: E
  print_operators: true # optional, default is false
  off_diagonal: true # optional, default is true
  order_by: operator # or score, or both, Optional, default is operator
  subtractvev: true # optional, default is true
  reweight: false # optional, default is false
  hermitian: false # optional, default is true
  sampling_mode: default # or jackknife, or bootstrap. Optional
  corrname: A Correlator! # optional, default is standard
  symbol_color: black # optional, default is blue
  symbol_type: square # optional, default is circle
  effective_energies:
    type: time_symmetric # optional, default is time_forward
    timestep: 2 # optional, default is 3
    plot_range: { tmin: 2.5, ymin: 0.0, tmax: 18.5, ymax: 1.25e-3 } # optional
  correlators:
    rescale: 3.5 # optional, default is 1.0
    plot_range: { tmin: 2.5, ymin: 0.0, tmax: 18.5, ymax: 1.25 } # optional
```

The `split_pdfs` key specifies whether the data should be split into multiple PDF files based on channel. The `print_operators` key specifies whether lists of operator strings should be printed to text files. The `off_diagonal` key specifies whether off diagonal correlator elements are shown. The `order_by` specifies how to order the correlators in the PDF file. If `subtractvev` is true, then VEV subtraction will be done for the cases where a non-zero VEV occurs. The `reweight` specifies whether the observables should be reweighted. The `sampling_mode` gives the resampling method to use. The `corrname`, `symbol_color`, and `symbol_type` keys are used to modify the plots (see `sigmond` documentation for more info). The `timestep` key gives the requested time step to use in the effective energy evaluation. The `plot_range` key gives the user a choice to specify the size of the correlator and effective energy plots. The `rescale` key is used to rescale the correlators by some rescale factor. If the value is `Default` (which is the default if the key is missing), then the resampling method specified in the `MCSamplingInfo` block is used.

2.8.4 Prune Operators

This task is used to compare different operator bases for the purpose of finding a final set of operators to use for a correlation matrix. The comparison will be written to a PDF file. The main information that will be given are the eigenvalues and condition number of A, $\widetilde{A}$, B, $\widetilde{B}$, $\widetilde{G}$, and $\widetilde{\widetilde{G}}$ (see appendix A). Further, it will also tell you whether the nullspace of B is entirely contained within the null space of A.

```
Prune:
  pivot_type: rolling_pivot # default is single_pivot
  improved_operators: true # default is false
  operator_bases:
    - name: op_basis_1 # optional
      operators: # optional if the basis already exists and the name is given
        - isodoublet S=1 P=(1,0,0) A1_1 kaon 2
        - kaon P=(1,0,0) A1_1 SS_2
        - isodoublet_kaon_pion A1_1 [P=(1,0,1) A1 LSD_1] [P=(0,0,-1) A2 TSD_2]
      rotation_times:
        - norm_time: 3
          metric_time: 11
          diag_time: 19
        - norm_time: 3
          metric_time: 15
```

```
        diag_time: 17
    - ...
    - isospin: doublet
      strangeness: 1
      momentum: [0,0,0]
      irrep: A1g
      irreprow: 1
  subtractvev: true # optional, default is true
  reweight: false # optional, default is false
  rotation_times:
    - norm_time: 3
      metric_time: 15
      diag_time: 21
    - norm_time: 3
      metric_time: 17
      diag_time: 23
  sampling_mode: default # or jackknife, or bootstrap. Optional
```

The `operator_bases` key specifies the different operator bases to consider. If both the `name` and `operators` keys are present in a given operator basis, then this particular operator basis will be stored for latter use based on the name given. If just the `name` key is given, then the operator basis stored with that name will be searched for. If the operator basis is not found, the basis will be skipped. If just the `operators` key is given, then the specified basis will be used but not stored for later use. Finally, if neither the `name` nor the `operators` keys are present, channel specification can be done to indicate that all operators found in that channel should be used.

If the `improved_operators` key is true, then configuration files will be produced that contain sections defining the improved operators from the different rotations. In order to define the matrices defined in appendix A the rotation times (τ_N, τ_0, τ_D) must be provided, which are are specified using the `rotation_times` key with a list of dictionaries. One can either specify a global set of rotation times to use for each operator basis and/or specify a set of rotation times for any given operator basis. The `sampling_mode` key works the same way as it did for the `ViewData` task.

2.8.5 Rotate Correlators

This task is for rotating correlation matrices (see appendix A).

```
RotateCorrelators:
  rotation_type: rolling_pivot # default is single_pivot
  rotate_by: bins # optional, default is samplings
  view_rotated_correlators: true # default is true
  min_time_sep: 2 # optional, if not present, smallest time separation used
  max_time_sep: 32 # optional, if not present, largest time separation used
  operator_bases:
    - name: op_basis_1
      operators: # optional if the basis already exists and the name is given
        - isodoublet S=1 P=(1,0,0) A1_1 kaon 2
        - kaon P=(1,0,0) A1_1 SS_2
        - isodoublet_kaon_pion A1_1 [P=(1,0,1) A1 LSD_1] [P=(0,0,-1) A2 TSD_2]
      rotation_times:
        - norm_time: 3
          metric_time: 11
          diag_time: 19
        - norm_time: 3
          metric_time: 15
          diag_time: 17
      max_cond_nums: [75]
    - ...
  rotation_times: # optional
    - norm_time: 3
      metric_time: 15
      diag_time: 21
    - norm_time: 3
      metric_time: 17
```

```
      diag_time: 23
  max_cond_nums: [50, 60, 100] # optional
  neg_eigen_alarm: -0.05 # optional, default is -0.01
  subtractvev: true # optional, default is true
  reweight: false # optional, default is false
  sampling_mode: default # or jackknife, or bootstrap. Optional
  corrname: A Correlator! # optional, default is standard
  symbol_color: black # optional, default is blue
  symbol_type: square # optional, default is circle
  effective_energies:
    type: time_symmetric # optional, default is time_forward
    timestep: 2 # optional, default is 3
    plot_range: { tmin: 2.5, ymin: 0.0, tmax: 18.5, ymax: 1.25e-3 } # optional
  correlators:
    rescale: 3.5 # optional, default is 1.0
    plot_range: { tmin: 2.5, ymin: 0.0, tmax: 18.5, ymax: 1.25 } # optional
```

The specification of the operator bases works in a similar manner as for the `Prune` task block. Some differences is the ability to specify a list of globally and operator basis dependent maximum condition numbers with the `max_condition_numbers` key (similar to how the `rotation_times` key works). Further, the `name` key, which is required, will be used to create a rotated basis, which can be referred to in later tasks. If the `view_rotated_correlators` key is true, then after the rotation has been done, a PDF will be produced that shows the results of the rotation. Note that the plotting keys are ignored if `view_rotated_correlators` is false, because no plots are produced in that case.

2.8.6 Fit Correlators

Next, we describe how to perform fits to correlators. There are three types of ways to perform these fits.

The possible values passed to `FitType` are: `1-exp`, `1-exp-sym`, `1-exp-const`, `1-exp-sym-const`, `2-exp`, `2-exp-sym`, `2-exp-const`, `2-exp-sym-const`, `geom`, and `geom-sym`. The `tmins` and `tmaxs` keys take a space separated list of t_{min} and t_{max} values. But, you can also specify a range of values as well. Now, the form of the actual `Fits` section is as follows

```
CorrelatorFits:
  minimizer_info:
    minimizer: minuit # or lmder or nl2sol, optional, default is lmder
    parameter_rel_tol: 1e-5 # optional, default is 1e-6
    chisquare_rel_tol: 1e-5 # optional, default is 1e-4
    max_iterations: 10000 # optional, default is 1024
    verbosity: high # or low or medium, optional, default is low
  fits:
    - name: 1_exp_fitname
      model: 1-exp
      tranges:
        - [3,20]
        - [4,20]
      tmins: [8, 10-15, 17]
      tmaxs: [20, 22-27]
      exclude_times: [10] # optional, default is empty (i.e. [])
      noise_cutoff: 1.4 # optional, default is 0.0 (i.e. no cutoff)
    - name: 2_exp_fitname
      model: 2-exp
      tmins: [3-15]
      tmaxs: [10-27]
      exclude_times: [12, 15] # optional, default is empty (i.e. [])
      noise_cutoff: 1.2 # optional, default is 0.0 (i.e. no cutoff)
  operators:
    - operator: kaon P=(0,0,0) A1g_1 SS_4
    - operator: pion P=(0,0,0) A1um_1 SS_3
      fits: # optional, if absent, then all fits are used
        - 1_exp_fitname
  operator_bases:
    - name: op_basis_1
```

```
      norm_time: 3  # optional
      metric_time: 11  # optional
      diag_time: 19  # optional
      max_cond_num: 150  # optional
      reference_energy: pion  # optional
      non_interacting_levels: pi_pi  # optional
      fits:  # optional, if absent, then all fits are used
        - 2_exp_fitname
  non_interacting_levels:
    - name: pi_pi
      levels:
        - [pion P=(0,0,0) A1um_1 SS_0, pion P=(0,0,0) A1um_1 SS_1]
        - [pion P=(0,0,1) A2m_1 SD_2, pion P=(0,0,-1) A2m_1 SS_4]
  reference_energies:
    - name: pion
      operator: pion P=(0,0,0) A1um_1 SS_0
      subtractvev: false # optional, default is false
      reweight: true # optional, default is false
      fit_model: 1-exp
      tmin: 15
      tmax: 27
      exclude_times: [18, 21] # optional, default is empty (i.e. [])
      noise_cutoff: 1.2 # optional, default is 0.0 (i.e. no cutoff)
  sampling_mode: default # or jackknife, or bootstrap. Optional
  cov_sampling_mode: default # or jackknife, or bootstrap. Optional
  subtractvev: true # optional, default is true
  reweight: false # optional, default is false
  effective_energies:
    type: time_symmetric
    timestep: 2 # optional, default is 3
    symbol_type: square # optional, default is circle
    symbol_color: black # optional, default is blue
    plot_range: { tmin: 2.5, ymin: 0.0, tmax: 18.5, ymax: 1.25e-3 } # optional
    show_approach: false # optional, default is true
    goodness: qual # optional, default is chisq
    corrname: A Correlator! # optional, default is standard
  tmin_plots:
    - operator: pion P=(0,0,1) A2m_1 SS_0
      tmin_range: 5-15
      tmax: 32
      plot_range: 0.0-1.25
```

For each channel passed to the `Channels` key, a fit will be performed to each diagonal correlator in that channel. For each operator passed to the `Operators` key, a fit will be performed to each diagonal correlator involving that operator. The `TimeStep` key is used for the effective energy plots. The `SamplingMode` key is used in the same way as for other tasks discussed above, and the `CovMatCalcSamplingMode` key is used to specify the sampling mode to use for cacluating the covariance matrix when the covariance on the full ensemble cannot be calculated (see section 10.2 "Extracting observables" from "Hadron Spectroscopy in Lattice QCD" notes). The value for the `Reference` key corresponds to a section name in which a fit is described. For example, For the `RatioFits` section, the value of the `NonInteractingCorrelators` corresponds to a section describing the non-interacting correlators to use for a ratio fit. In this case, you must map each energy level to a set of operators corresponding to correlators that will closely represent the non-interacting levels. For example, The `TminPlots` allows you to specify how a t_{min} plot should be made. If this key is missing, then no t_{min} plots will be made. The values for the `TminPlots` key correspond to sections of the form

2.8.7 Fit Anisotropy

The form of the block for an anisotropy fit is

```
AnisotropyFit:
  MinimizerInfo:
    Method: Minuit2 # or LMDer or Minuit2NoGradient, optional, default is LMDer
```

```
    ParameterRelTol: 1e-5 # optional, default is 1e-6
    ChiSquareRelTol: 1e-5 # optional, default is 1e-4
    MaximumIterations: 10000 # optional, default is 1024
    Verbosity: High # or Low or Medium, optional, default is Low
  Energies:
    - MomentumSquared: 0
      Operator: pion P=(0,0,0) A1um_1 SS_0
      FitModel: 1-exp
      trange: 17-27
      ExcludeTimes: 21 # optional, default is empty
      LargeTimeNoiseCutoff: 1.4 # optional, default is 0.0 (i.e. no cutoff)
      SubtractVEV: false # optional, default is true
      Reweight: true # optional, default is false
    - MomentumSquared: 1
      Operator: pion P=(0,0,1) A2m_1 SS_0
      FitModel: 1-exp
      trange: 18-28
      ExcludeTimes: 19 # optional, default is empty
      LargeTimeNoiseCutoff: 1.1 # optional, default is 0.0 (i.e. no cutoff)
      SubtractVEV: false # optional, default is true
      Reweight: true # optional, default is false
    - MomentumSquared: 2
      Operator: pion P=(0,1,1) A2m_1 SS_0
      FitModel: 1-exp
      trange: 15-27
      ExcludeTimes: [16, 21] # optional, default is empty
      LargeTimeNoiseCutoff: 1.2 # optional, default is 0.0 (i.e. no cutoff)
      SubtractVEV: false # optional, default is true
      Reweight: true # optional, default is false
    - MomentumSquared: 3
      Operator: pion P=(1,1,1) A2m_1 SS_0
      FitModel: 1-exp
      trange: 17-23
      ExcludeTimes: 21 # optional, default is empty
      LargeTimeNoiseCutoff: 1.2 # optional, default is 0.0 (i.e. no cutoff)
      SubtractVEV: false # optional, default is true
      Reweight: true # optional, default is false
  Plot:
    ParticleName: pion # optional
    SymbolColor: black # optional, default is blue
    SymbolType: square # optional, default is circle
    Goodness: qual # optional, default is chisq
    PlotRange: { Pmin: -0.5, Emin: 0.0, Pmax: 4.5, Emax: 0.75 } # optional
```

2.8.8 Find Spectrum

This task is for choosing the fit values for each energy level in a particular channel. A PDF file will be produced to show the spectrum.

```
Spectrum:
  SamplingMode: default # or jackknife, or bootstrap. Optional
  Channels:
    - OperatorBasis: A1_P1_Nops4
      RotationTime: (3,15,21)
      MaxConditionNumber: 100
      References:
        - Operator: pion P=(0,0,0) A1um_1 SS_0
          SubtractVEV: false # optional, default is true
          Reweight: true # optional, default is false
          FitModel: 1-exp
          trange: 15-27
          ExcludeTimes: 18 21 # optional, default is empty
          LargeTimeNoiseCutoff: 1.2 # optional, default is 0.0 (i.e. no cutoff)
      Levels:
        - Level: 0
          FitModel: 1-exp
          trange: 17-23
          ExcludeTimes: 21 # optional, default is empty
          LargeTimeNoiseCutoff: 1.2 # optional, default is 0.0 (i.e. no cutoff)
```

```
SubtractVEV: false # optional, default is true
Reweight: true # optional, default is false
TminPlot:
  - Operator: pion P=(0,0,1) A2m_1 SS_0
    TminRange: 5-15
    Tmax: 32
    PlotXRange: 3,23 # optional
    PlotYRange: 0.0,0.75 # optional
```

I would also like to include some control over how other plots are made.

2.9 Optional Arguments

There also exist two possible optional arguments that can be placed in the input file

```
precision: 3                # default is 2
latex_compiler: lualatex    # default is based on pylatex library
```

3 Examples

Some examples...

A Breif Review of Correlator Analysis

Let the raw $N \times N$ correlator matrix be denoted $\mathcal{C}(t)$. An early time sepration τ_N is used to rescale the raw correlator matrix. The rescaled correlator matrix is defined by

$$C_{ij}(t) = \frac{\mathcal{C}_{ij}(t)}{\left(\mathcal{C}_{ii}(\tau_N)\mathcal{C}_{jj}(\tau_N)\right)^{1/2}}. \tag{1}$$

Next, we introduce the matrices $A \equiv C(\tau_D)$ and $B \equiv C(\tau_0)$, where $\tau_N \leq \tau_0 < \tau_D$. The eigenvectors corresponding to the $N_0 \leq N$ largest eigenvalues of B such that $\frac{|\lambda^N|}{|\lambda^{N-N_0+1}|} \leq \xi_{\max}$, where λ^i is the ith eigenvalue of B ordered from smallest to largest and $\xi_{\max}$ is the largest acceptable condition number, are put into a matrix P_0. Then, the matrices $\widetilde{A} \equiv P_0^\dagger A P_0$ and $\widetilde{B} \equiv P_0^\dagger B P_0$ are defined. Finally, the matrix $\widetilde{G} \equiv \widetilde{B}^{-1/2}\widetilde{A}\widetilde{B}^{-1/2}$ is diagonalized. The eigenvectors of $\widetilde{G}$ are then used to perform rotations at other times, where another SVD drop is done as was done for B. Denote this final matrix $\widetilde{\widetilde{G}}$.