7/23 Meeting

Christina

Three Global Methods

1. Monte Carlo (what we are currently using)
2. Differential Evolution
3. Basinhopping (a markov chain Monte Carlo method)

Monte Carlo

Advantage: takes little time (although it becomes significant when the data set becomes huge)

Disadvantage: not robust to noise

Differential Evolution

The algorithm randomly selects n points (population) from the 6D space (with user-defined range), then recombines the points and updates the points' new positions until there is only one point with the lowest chi^2 value left (or that the number of iteration exceeds the limit)

Advantage: very robust to noise; large rate of convergence when the population size is large

Disadvantage: Takes a significantly longer time (>30 seconds per event)

Basinhopping

Given a starting point (initial condition), the algorithm calculates the local minimum of chi^2 value using a user-defined local method, then applies a random perturbation to the coordinates. The new position is accepted /rejected based on the Metropolis criteria. The point with the lowest chi^2 value is updated as the newest global minimum

Advantage: less time than Differential Evolution but gives comparable results

Disadvantage: not robust to noise

Event 504

Data with noise

Monte-Carlo: 90.04

Basinhopping with SLSQP as the local method: 83.8

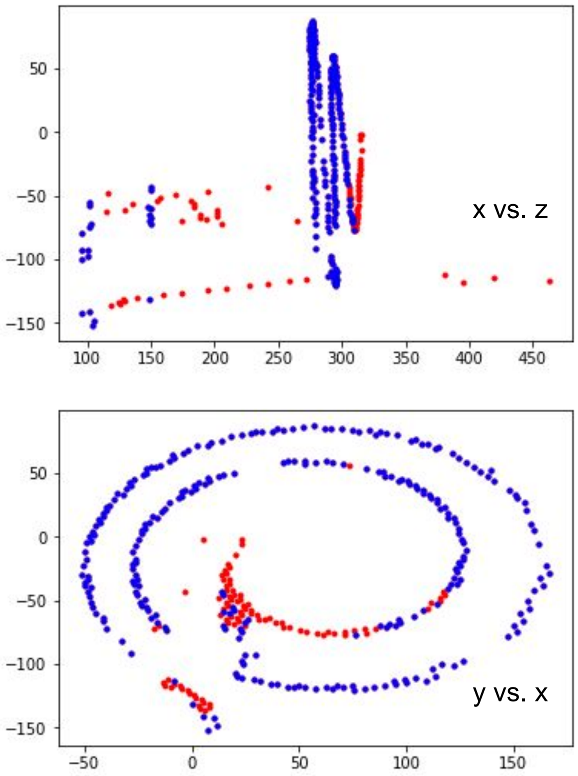
Differential Evolution: 68.06

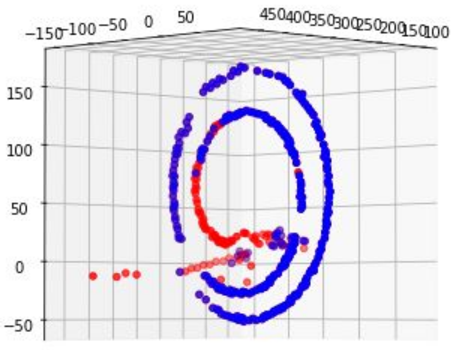
Data with reduced noise

Monte-Carlo: 45.0

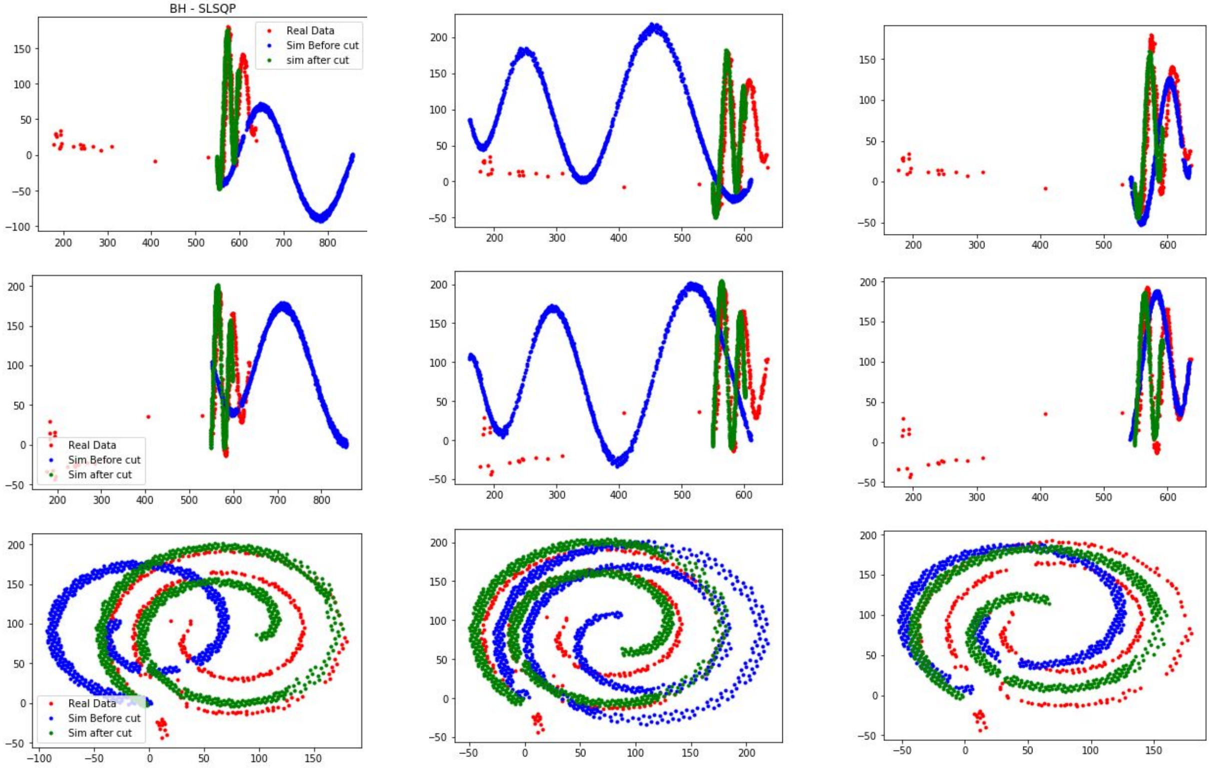
Basinhopping with SLSQP as the local method: 42.7

Differential Evolution: 57.61





blue: after cleaning
red: before cleaning



x vs. z

y vs. z

y vs. x

Basinhopping | Monte Carlo | Diff Evolution

Red: real data (with noise)
Blue: fittings of data with noise
Green: fitting of data without noise

Event 765

Data with noise

Monte-Carlo: 100.05

Basinhopping with SLSQP as the local method: 95.8

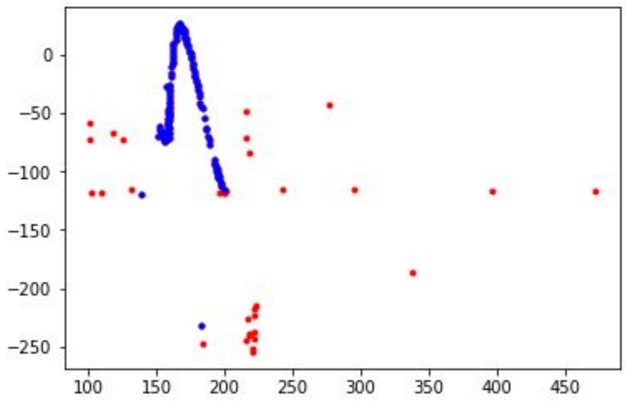
Differential Evolution: 35.45

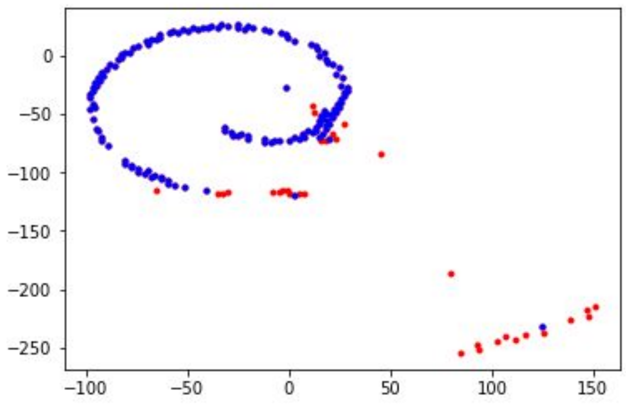
Data with reduced noise

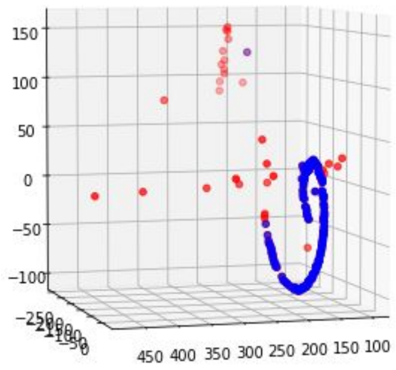
Monte-Carlo: 33.53

Basinhopping with SLSQP as the local method: 37.9

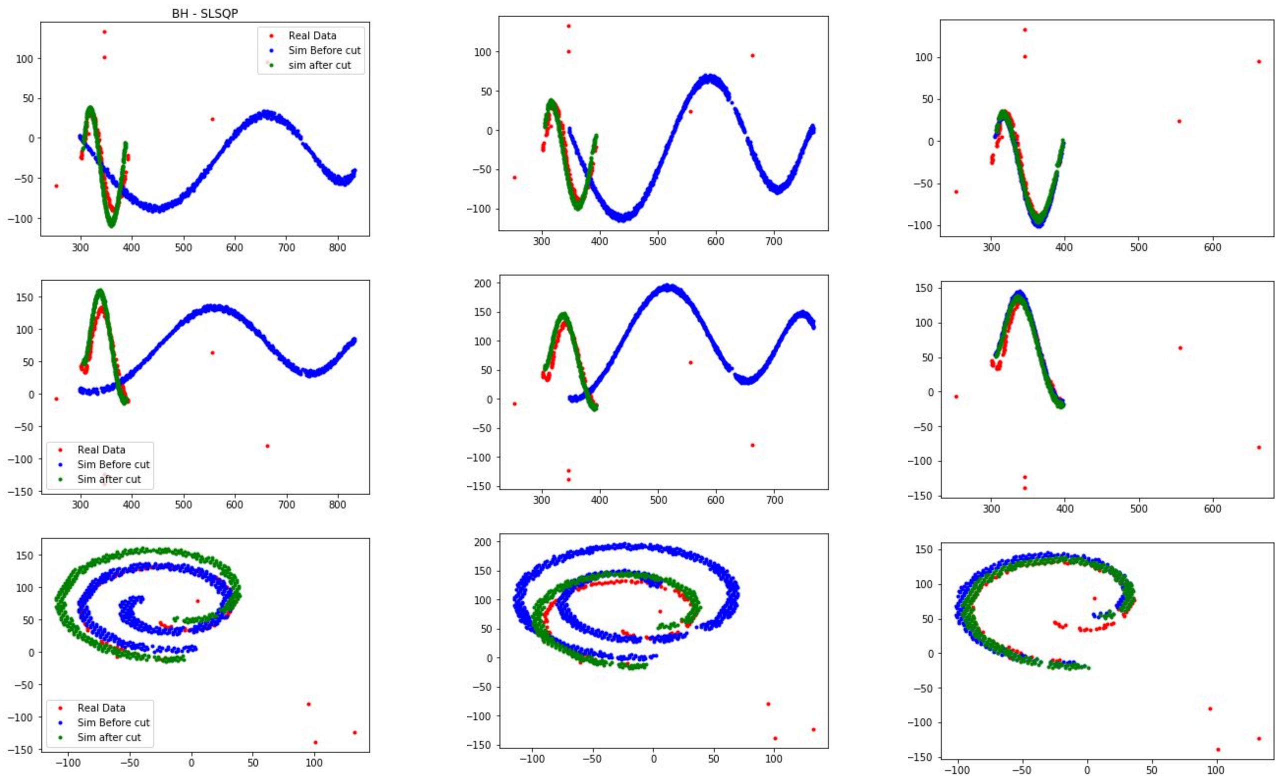
Differential Evolution: 34.27







blue: after cleaning
red: before cleaning



x vs. z

y vs. z

y vs. x

Basinhopping | Monte Carlo | Diff Evolution

Red: real data (with noise)
Blue: fittings of data with noise
Green: fitting of data without noise

Monte Carlo

```
average time with noise: 0.9150381401965493 seconds
average time without noise: 0.7961778515263608 seconds
average chi2 value (with noise): 58.50360822485877
average chi2 value (without noise): 35.912268307746764
```

Differential Evolution

```
average time: 33.156591317483354 seconds
average chi2 value (with noise): 42.02732836605447
average chi2 value (without noise): 42.00646858265847
```

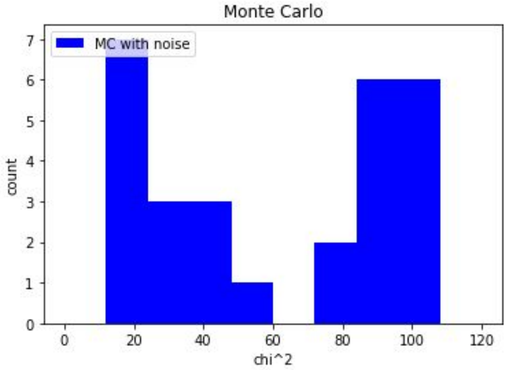
Basinhopping
niter = 10

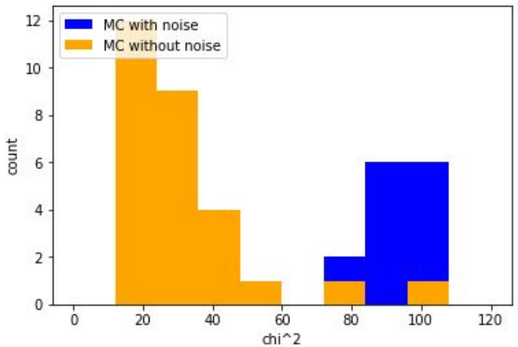
```
average time for Nelder-Mead: 7.100889333656856 seconds
chi2 value (with noise) for Nelder-Mead: 56.602099562368004
chi2 value (without noise) for Nelder-Mead: 36.739391372534904
average time for Powell: 15.809054085186549 seconds
chi2 value (with noise) for Powell: 72.39768213081769
chi2 value (without noise) for Powell: 48.439384824847956
average time for CG: 5.473999002150127 seconds
chi2 value (with noise) for CG: 85.06981919655972
chi2 value (without noise) for CG: 68.7291566328876
average time for BFGS: 6.69640759059361 seconds
chi2 value (with noise) for BFGS: 71.78550302526396
chi2 value (without noise) for BFGS: 57.42162601972965
average time for L-BFGS-B: 5.841121426650456 seconds
chi2 value (with noise) for L-BFGS-B: 81.43751391791639
chi2 value (without noise) for L-BFGS-B: 66.45097907673568
average time for TNC: 6.563905213560377 seconds
chi2 value (with noise) for TNC: 74.96609741576006
chi2 value (without noise) for TNC: 58.198438073015446
average time for COBYLA: 1.666680829865592 seconds
chi2 value (with noise) for COBYLA: 78.49874464606329
chi2 value (without noise) for COBYLA: 66.19309393961564
average time for SLSQP: 2.873416449342455 seconds
chi2 value (with noise) for SLSQP: 62.756947144902846
chi2 value (without noise) for SLSQP: 37.88723205549305
```

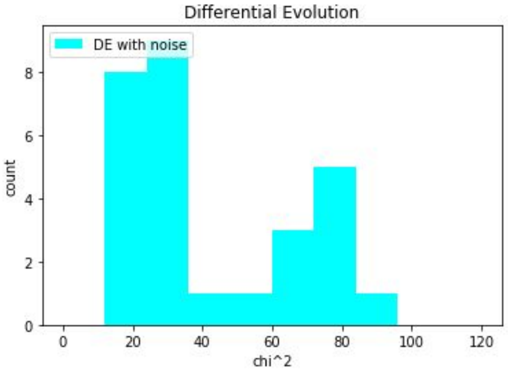
Basinhopping
niter = 25

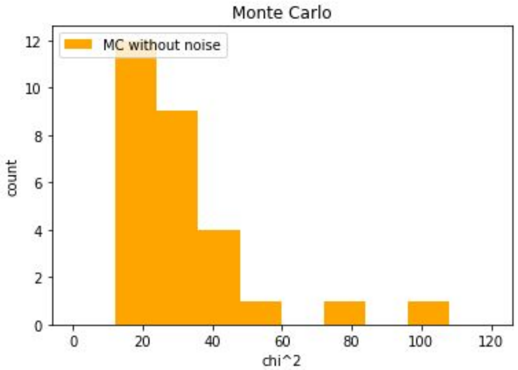
```
average time for Nelder-Mead: 20.268136343785695 seconds
chi2 value (with noise) for Nelder-Mead: 49.02851080971168
chi2 value (without noise) for Nelder-Mead: 30.06558285392754
average time for Powell: 36.70845403841564 seconds
chi2 value (with noise) for Powell: 59.04416700450565
chi2 value (without noise) for Powell: 47.49919979627662
average time for BFGS: 16.16957257475172 seconds
chi2 value (with noise) for BFGS: 68.93500843710063
chi2 value (without noise) for BFGS: 46.31901726906612
average time for SLSQP: 6.993133834430149 seconds
chi2 value (with noise) for SLSQP: 63.31442266173478
chi2 value (without noise) for SLSQP: 33.89991591501348
```

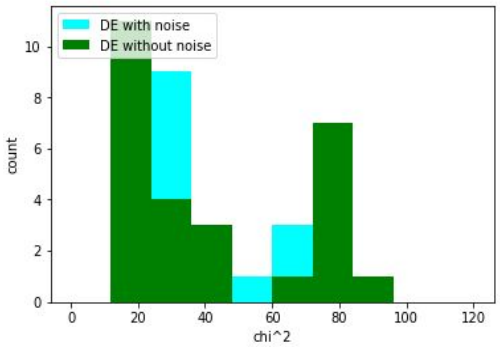
Histograms

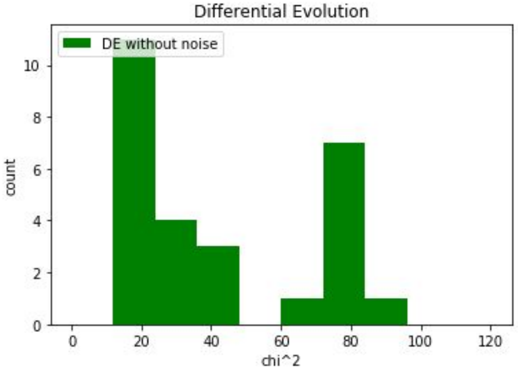

Monte Carlo
MC with noise
count
chi^2

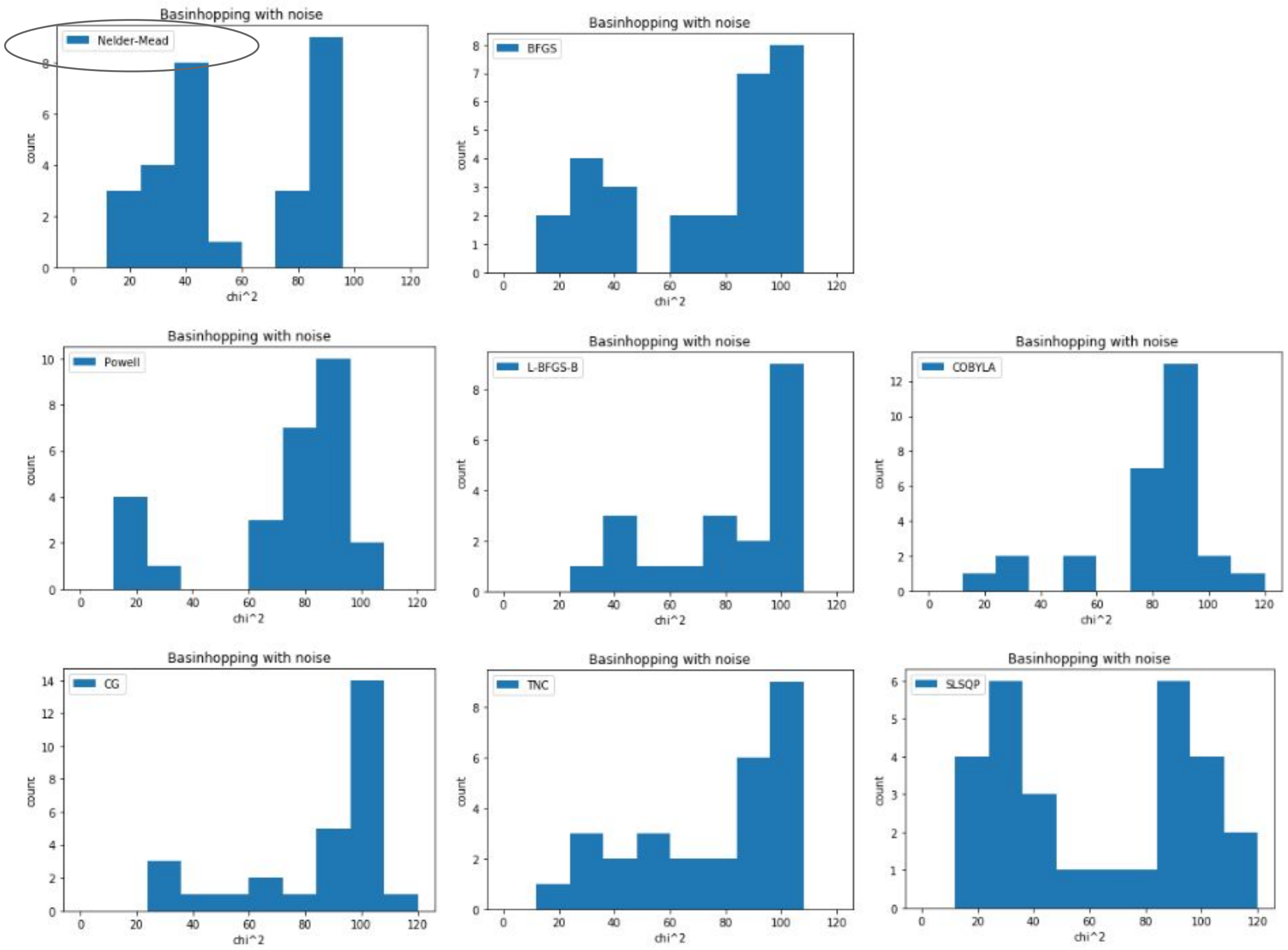

MC with noise
MC without noise
count
chi^2

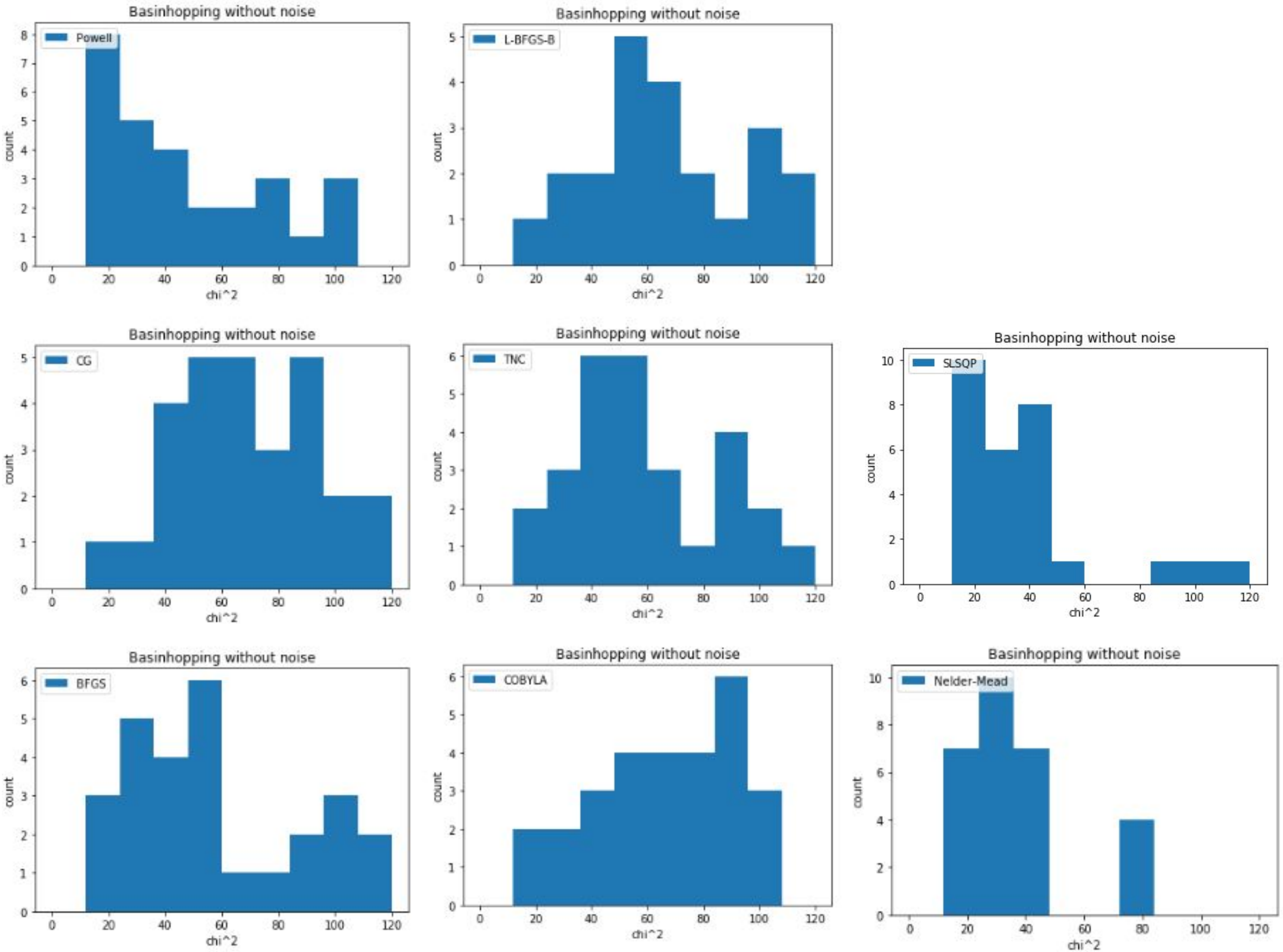

Differential Evolution
DE with noise
count
chi^2

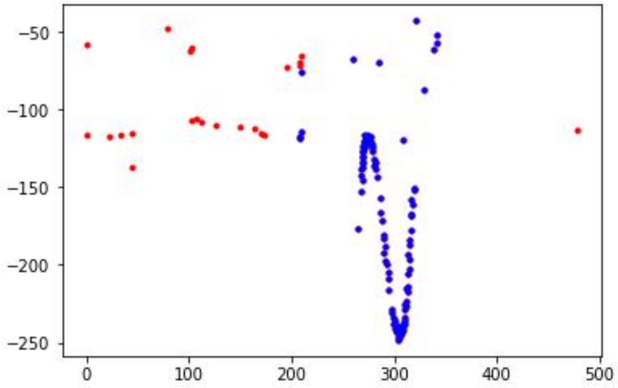

Monte Carlo
MC without noise
count
chi^2

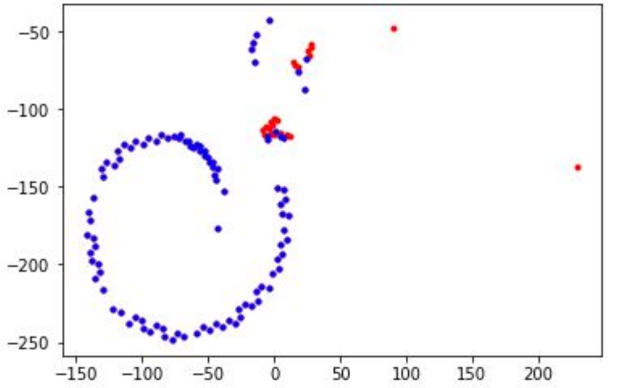

DE with noise
DE without noise
count
chi^2


Differential Evolution
DE without noise
count
chi^2


Basinhopping with noise
Nelder-Mead
BFGS
Powell
L-BFGS-B
COBYLA
CG
TNC
SLSQP
count
chi^2


Basinhopping without noise
Powell
Basinhopping without noise
L-BFGS-B
Basinhopping without noise
CG
Basinhopping without noise
TNC
Basinhopping without noise
SLSQP
Basinhopping without noise
BFGS
Basinhopping without noise
COBYLA
Basinhopping without noise
Nelder-Mead
count
chi^2



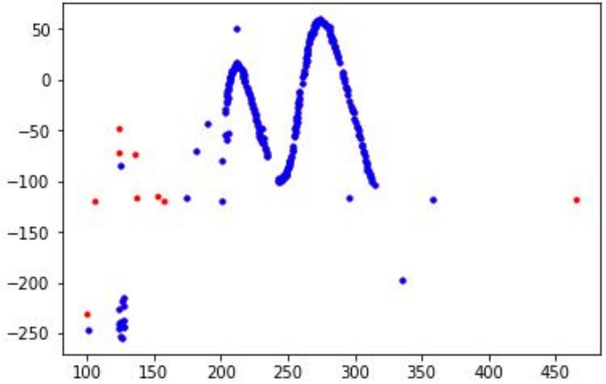


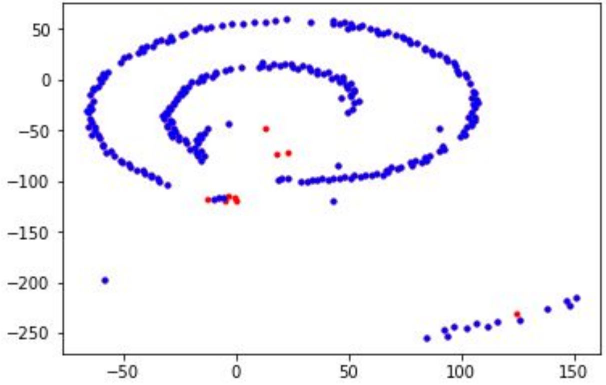
Cleaning's effect on processing time (Monte Carlo)

blue: after cleaning
red: before cleaning

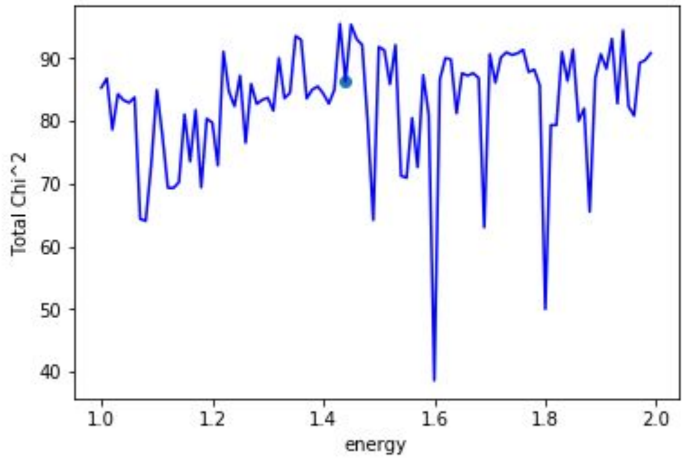
We can see that the cleaned data takes a lot less time

```
Monte Carlo event 12 with noise: 0.6670424938201904 seconds
position chi2: 12.25689314692066 energy chi2: 10.461460892477122 vertex chi^2: 2.2916093421940524 total chi2: 25.00996338
1591835
Monte Carlo event 12 without noise: 0.37213635444641113 seconds
position chi2: 6.920917079017351 energy chi2: 17.256624086433263 vertex chi^2: 2.6925715154967915 total chi2: 26.87011268
0947404
```

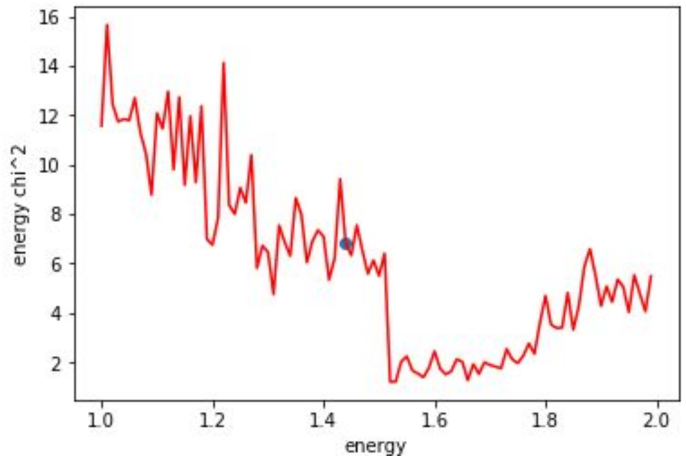





```
Monte Carlo event 9 with noise: 0.9181435108184814 seconds
position chi2: 16.211907888359203 energy chi2: 3.762711409690692 vertex chi^2: 5.841481496590662 total chi2: 25.816100794
640555
Monte Carlo event 9 without noise: 0.5298173427581787 seconds
position chi2: 15.814149343876108 energy chi2: 3.474156675673004 vertex chi^2: 6.084244782769689 total chi2: 25.372550802
3188
```




```
Text(0,0.5,'energy chi^2')
```



The energy chi^2 only counts towards a small part of the total chi^2 value.

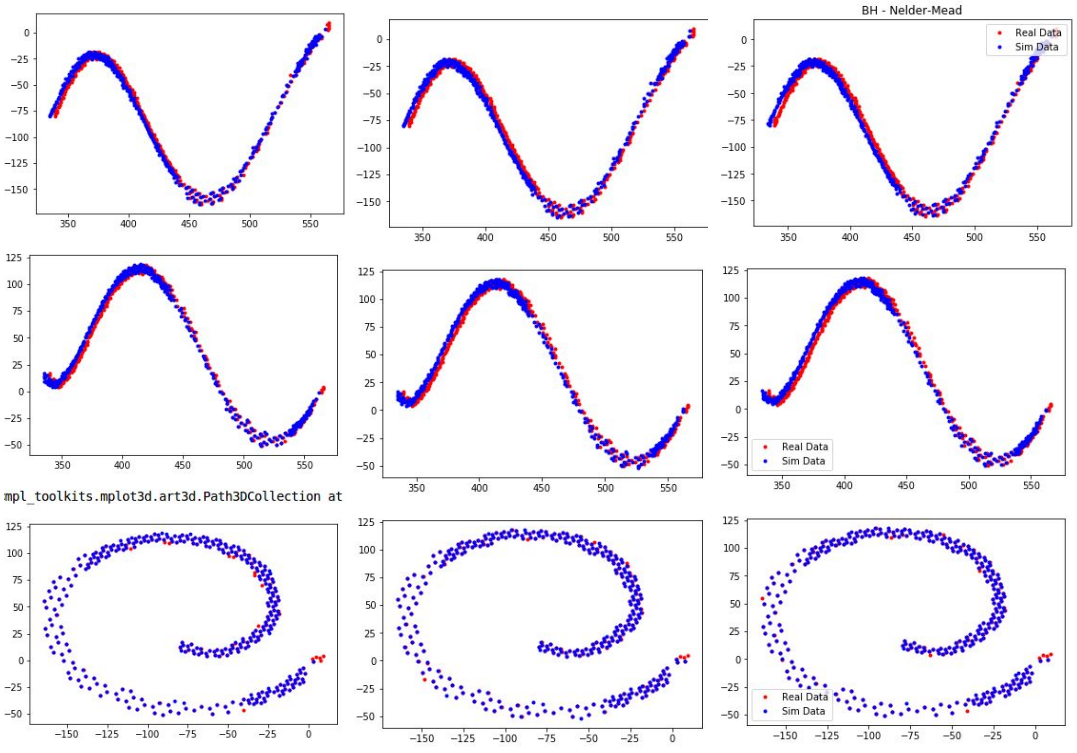
The blue points here shows the energy value that minimizes the total value of the objective function.

Scipy Reference Guide

Local methods: scipy.optimize.minimize
https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.minimize.html#scipy.optimize.minimize
Global methods:
Basinhopping:https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.basinhopping.html#scipy.optimize.basinhopping
Differential Evolution:https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.differential_evolution.html#scipy.optimize.differential_evolution
https://en.wikipedia.org/wiki/Differential_evolution

Simulated Data - proton-like event

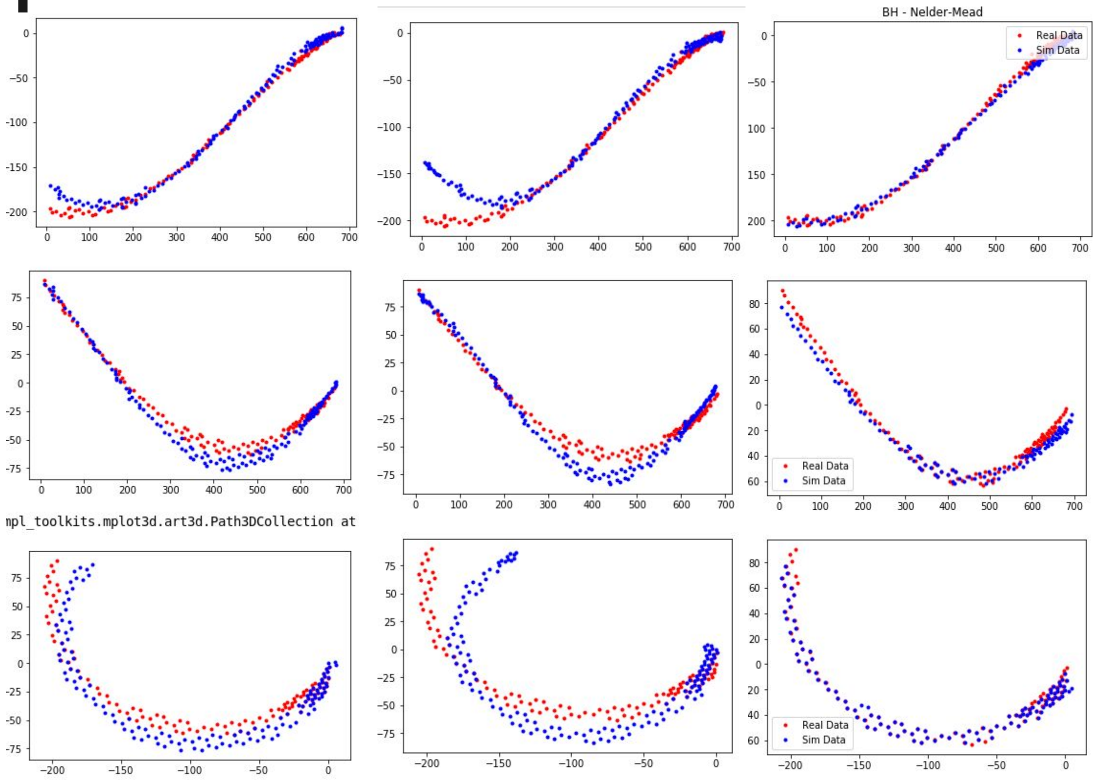
Monte-Carlo: 1.59
Differential Evolution: 1.26
Basinhopping: 1.78 with Nelder-Mead
(SLSQP did not work for simulated data)



Monte Carlo | Differential Evolution | Basinhopping

Simulated Data - less proton-like

Monte-Carlo: 30.899
Differential Evolution: 40.335
Basinhopping: 14.07 with Nelder-Mead
18.27 with SLSQP



Monte Carlo

Differential Evolution

Basinhopping