

Add AD into analytic continuation algorithm of ACFlow

Current process and challenges

Kaiwen Jin

HKUST-GZ

2025-01-22

Challenges we meet

1. AD on AAA is numerically instable and AAA may be undifferentiable;
2. Stochastic method works so bad. Should we give up or improve them?
3. Can we know poles of $A(w)$ in advance? Otherwise We have no effective method on discrete $A(w)$.

Contents

- 0. Conclusion.
- 1. Continuous spectral density function ($A(w)$ for short)
 - 1.1 Features and improvement of Barycentric/AAA algorithm and challenges of it's AD.
 - 1.2 The implementation of the AD of chi2kink (a MaxEnt method).
 - 1.3 Awful performance of stochastic methods.
- 2. Discrete $A(w)$
 - 2.1 Do we know poles of $A(w)$ in advance? And limitation of directly solve amplitudes.
 - 2.2 Effectiveness and limitation of Bary and chi2kink.

2.3 Awful performance of statistic methods.

3. What should we do about stochastic methods?

0. Conclusion

1. Accuracy and noise robustness: Chi2kink(MaxEnt) >> math methods(AAA,Nanv) >> Stochastic method
2. Speed : Math methods > Chi2kink >> Stochastic method
3. Add AD:

Chi2kink : well done

AAA : essentially difficult

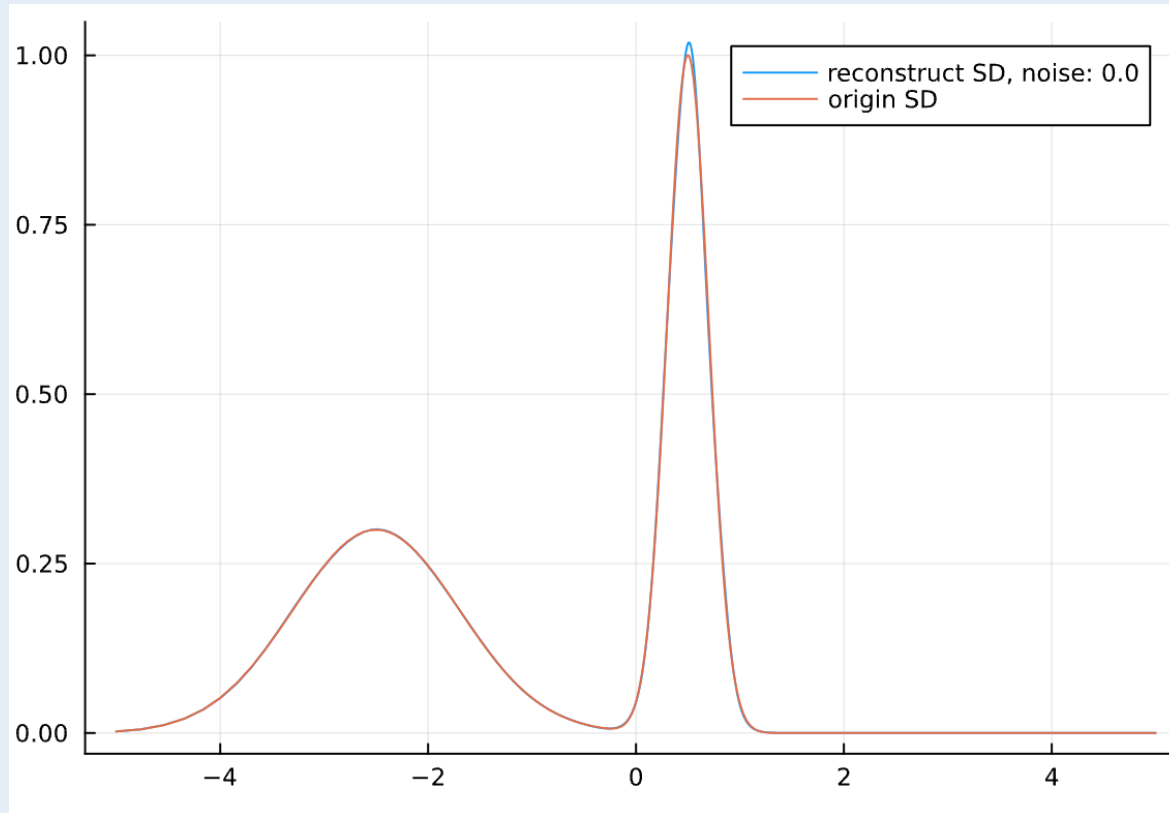
Stochastic Methods : unknown
4. Challenges we meet :

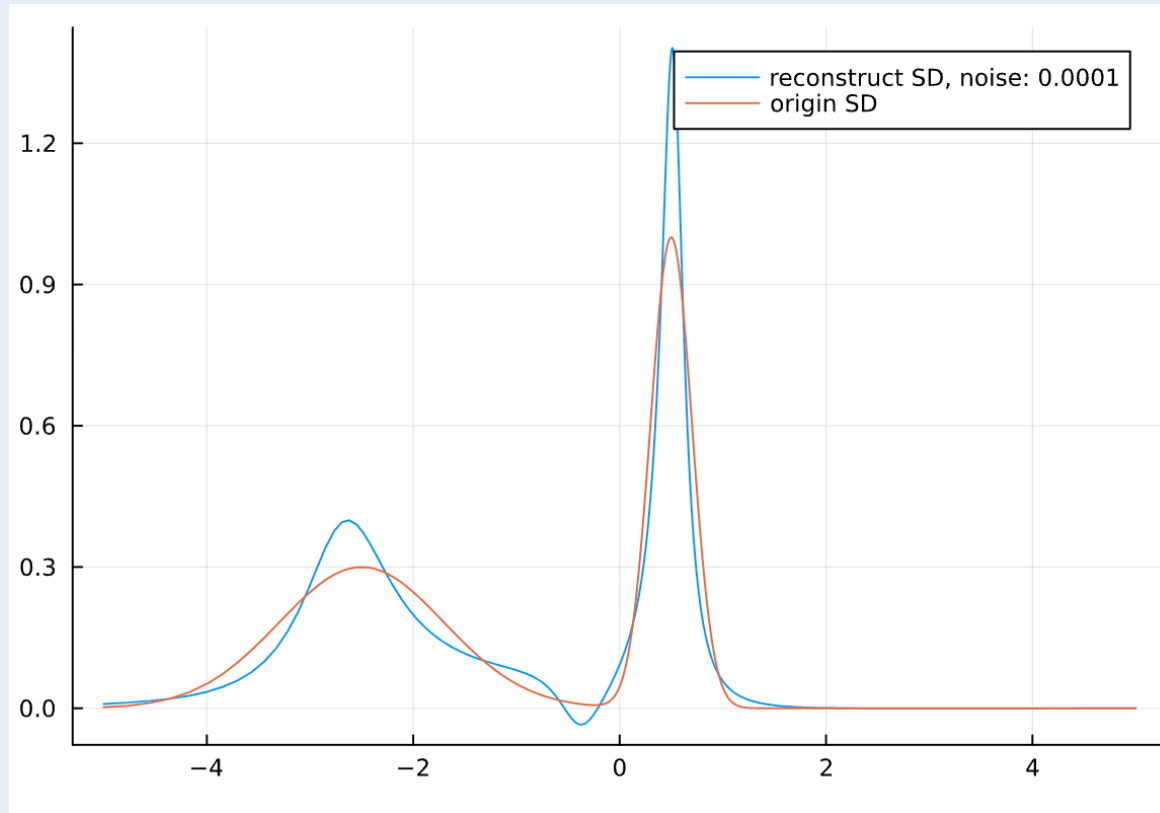
- (1) AD on AAA is numerically instable and AAA may be undifferentiable;
- (2) Stochastic method works so bad. Should we give up or improve them?
- (3) Can we know poles of $A(w)$ in advance? Otherwise We have no effective method on discrete $A(w)$. Of course I see most algorithm assume we don't know poles in advance, but knowing poles in advance is the only effective method I can think out for discrete $A(w)$.

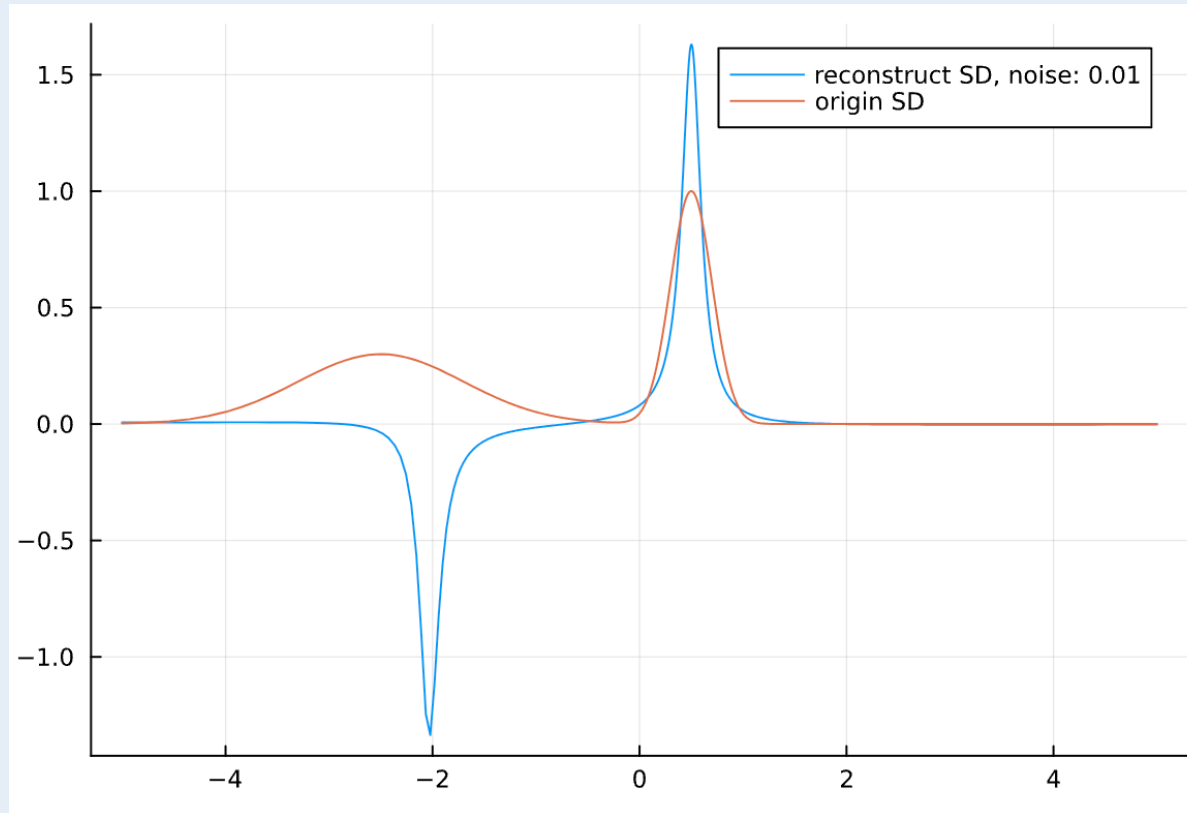
1. Continuous $A(w)$

1.1 Features and improvement of Barycentric/AAA algorithm and challenges of it's AD.

Calling it barycentric algoorithm, what we use in fact is aaa algorithm. AAA and Nevanlinna are both purely mathematical algorithms. They are fast but extremely instable to noise. AAA for example:







ACflow's wrong use of svd in LinearAlgebra will cause the algorithm go wrong. I have fixed the bug.

The main problem of it's AD is, in AAA we can't avoid applying svd on a Lowner matrix

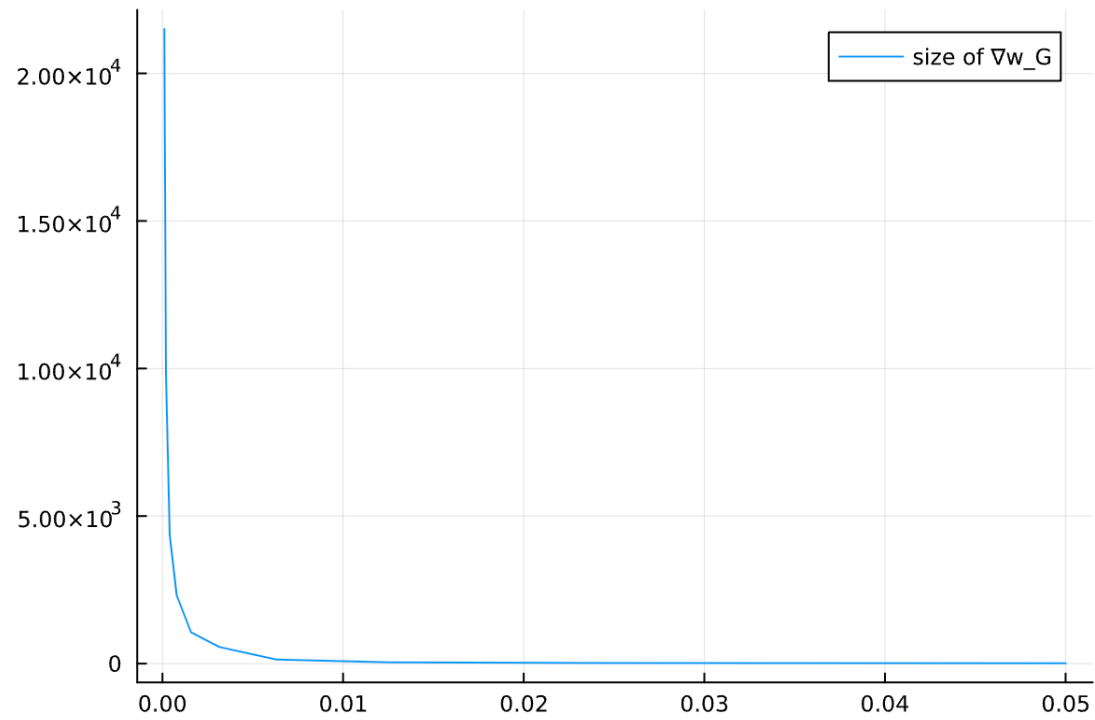
$$C = \left(\frac{G(iw_i) - G(iw_j)}{iw_i - iw_j} \right)_{i \times j}$$

and Cauchy matrix

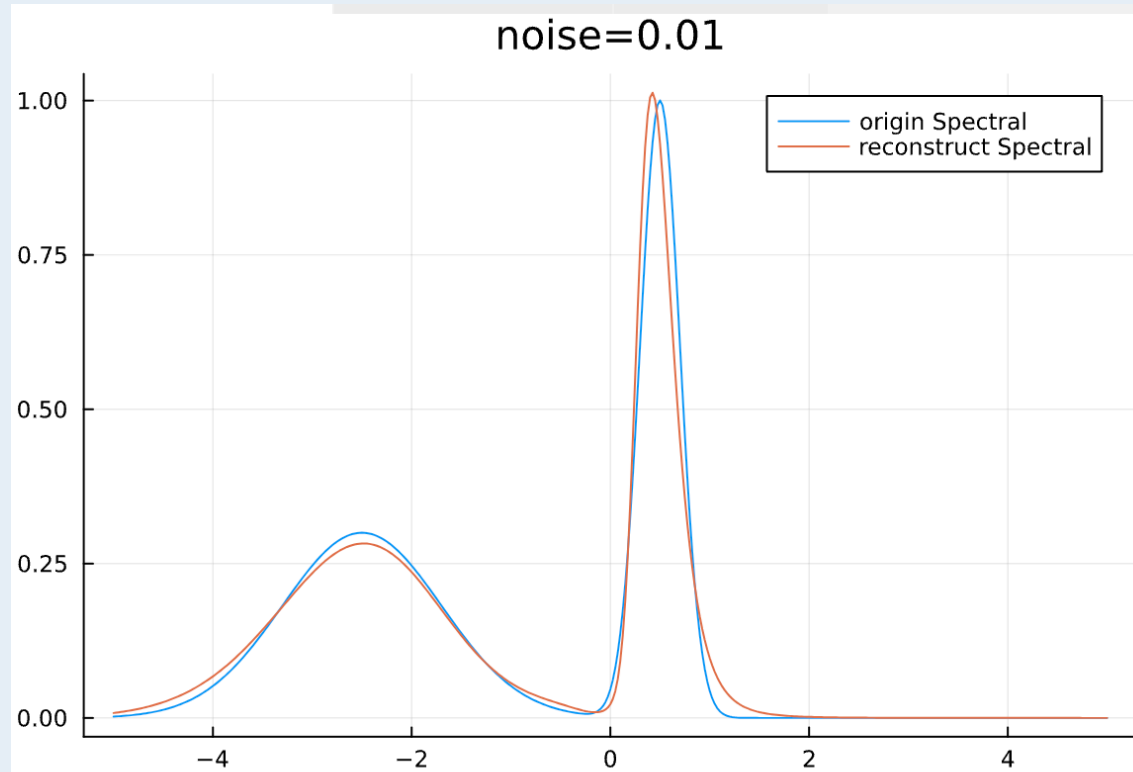
$$L = \left(\frac{1}{iw_i - iw_j} \right)_{i \times j}$$

Both of them are ill-conditioned matrices and will cause great numerical instability. Derivative formula of svd falls here. Some term of the formula should be zeros in theory but it ends to be more than $1e5$ in practical code running.

Finite difference (FD for short) only works in rare specific cases. In most cases, AD and FD just blow up. And some FD results just suggest this method may be just undifferentiable. The following chart shows the partial derivative of weights coefficient got in AAA with respect to G we get:



1.2 The implementation of the AD of chi2kink (a MaxEnt method) It's fast and has extremely strong noisy robustness. But it only work for continous $A(w)$. The following chart is its performance under $1e-2$ noise:



Now we have realized AD of chi2kind with combination of formula and Zygote. The running result is reasonable.

xamples >  MyChi2kink.jl > ...

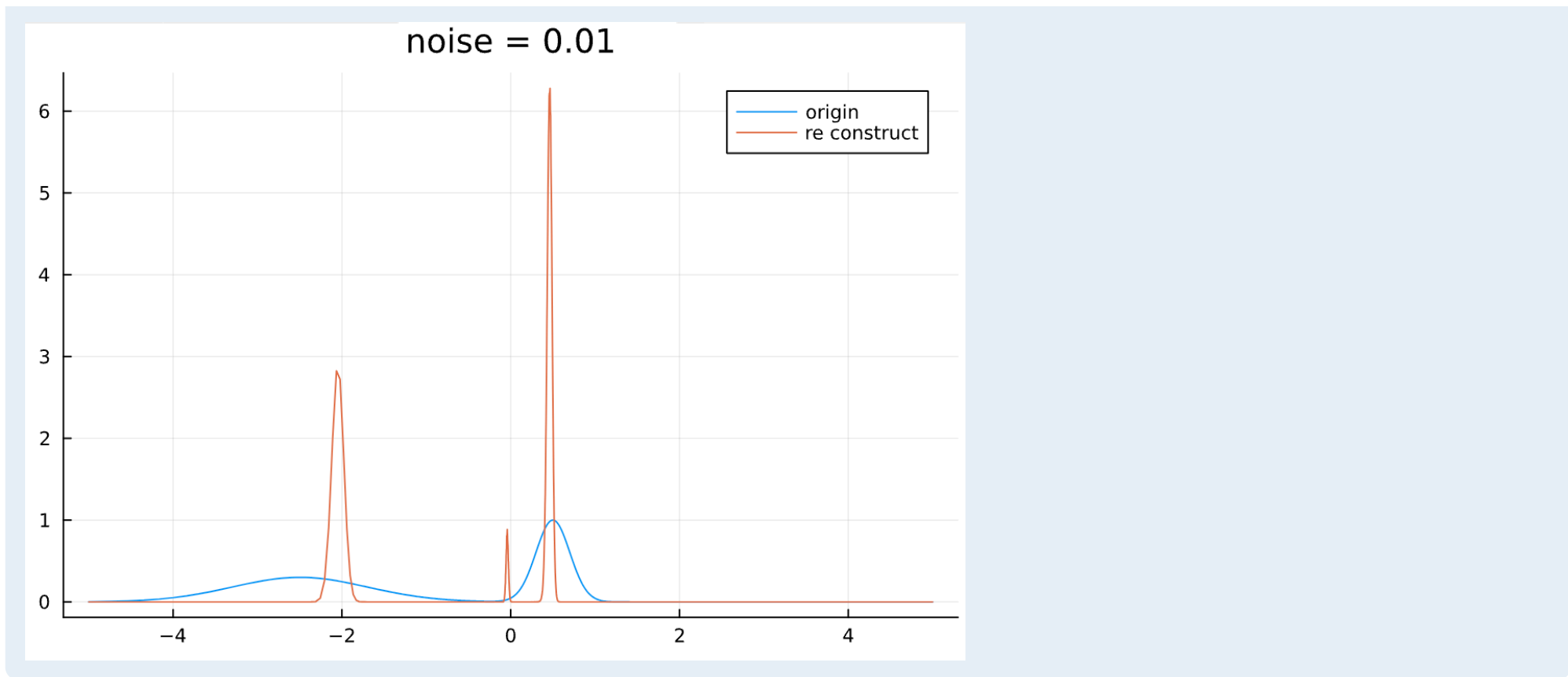
```

1  using ACFlowSensitivity [✓]
2  using Plots,Zygote,LinearAlgebra [✓]
3
4  μ=[0.5,-2.5];σ=[0.2,0.8];peak=[1.0,0.3]; [✓]
5  A=continous_spectral_density(μ,σ,peak); [✓]
6  β=10.0; [✓]
7  N=20; [✓]
8  output_bound=5.0; [✓]
9  output_number=401; [✓]
10 noise=0.0; [✓]
11 Gvalue=generate_G_values_cont(β,N,A;noise=noise); [✓]
12 output_range=range(-output_bound,output_bound,output_number); [✓]
13 output_range=collect(output_range); [✓]
14 iwn=(collect(0:N-1).+0.5)*2π/β * im; [✓]
15
16 Aout=my_chi2kink(iwn,Gvalue,output_range) [ 401-element Vector{Float64}:
17 plot(output_range,A.(output_range),label="origin Spectral ",title="noise=$noise") [ Plot{Plots.GRBack
18 plot!(output_range,Aout,label="reconstruct Spectral") [ Plot{Plots.GRBackend() n=2}
19
20 ADAout = ADchi2kink(iwn,Gvalue,output_range) [ 20-element Vector{ComplexF64}:
21 @show norm(ADAout) [ 69.51897796595368
22
23

```

1.3 Awful performance of stochastic methods.

AAA need second and Chi2kink need less than half a minute. But Sandvic algorithm need at least 15 minutes and its result is extremely awful:



.

According to ACFlow's document, other stochastic methods are equally slow or even much more. And accuracy of stochastic methods is worse than Barycentric/AAA.

2. Discrete $A(w)$

2.1 Do we know poles of $A(w)$ in advance? And limitation of directly solve amplitudes.

If we know accurate poles in advance, we can just solve

$$\left(\frac{1}{iw_i - p_j} \right)_{i \times j} X = (G(iw_i))_{i \times 1}$$

Limitation is that the cauchy matrix $\left(\frac{1}{iw_i - p_j} \right)_{i \times j}$ is an ill-conditioned matrix. But we have existing method to solve this question.

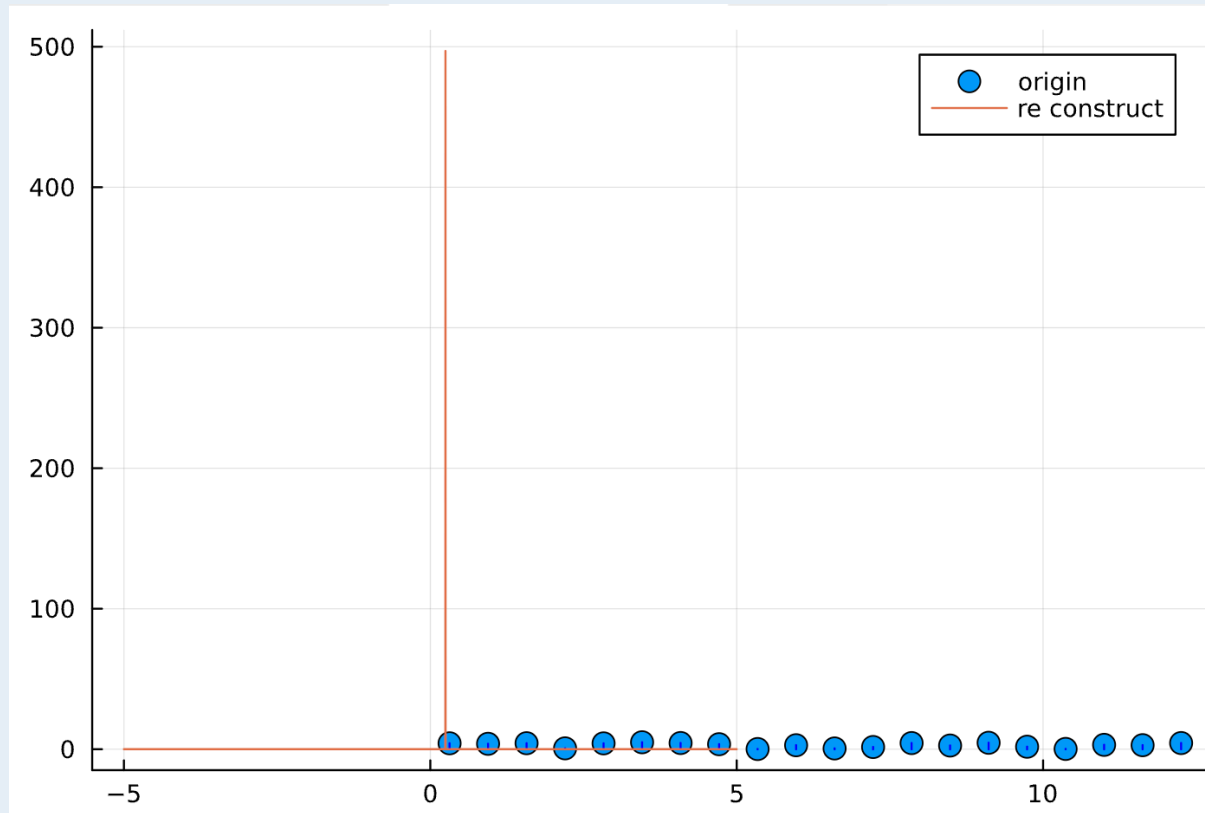
The following discussion is based on that we don't know w_n in advance.

2.2 Effectiveness and limitation of Bary and chi2kink.

Chi2kink, or all maximum entropy methods, is not suitable for discrete $A(w)$.

Barycentric/AAA improved by me can find first several poles and amplitudes relatively accurately. But for bigger w_n the result just blows up. So it can't be applied in practice.

2.3 Awful performance of statistic methods. ACFlow doesn't give discrete example. I test the algorithm by example I write so I am not sure if I write an example not suitable. ACFlow documents claim these stochastic methods' accuracy is even worse than Barycentric/AAA. in my example, it is so:



It only find the first pole, which is same as original AAA (not improved by me).
And it's still extremely slow.

What should we do about stochastic methods?

With stochastic algoorithm's awful performance, do we need its AD?

If we know poles in advance, it's tatolly different question. If not so and we have to add AD to stochastic algorithm, I have some ideas about future direction:

1. Just add AD to stochastic algoorithm of ACFlow.
2. Rewrite and improve ACFlow's code realization. If it work well (at least for discrete $A(w)$), we add AD.

3. Versions of stochastic algorithms that ACFlow realize is relatively old. The Sandvic algoorithm it realizes was published in 2017 and a improved version was published in 2020. We can find new version and realize it to see if they work better.