

Tester LSDE

October 27, 2020

1 Tester: Linear SDE's

A Linear SDE for a n -dimensional system is of the form

$$dX_t = AX_t dt + \sigma dW_t$$

where W_t is a m -dimensional Wiener process.

```
[21]: using PyPlot
      using Random

      # Get software to generate model
      include("../LinearSDE/modgen_LSDE.jl")

      # Get model reduction software being tested
      include("../Tools/Model_Reduction_Dev.jl")

      # Get tester tools
      include("testertools.jl")
```

```
[21]: causal_test (generic function with 1 method)
```

1.1 1-D

```
[2]: # Model run Parameters
      t_start = 0
      t_stop  = 1e3
      h       = 1e-2

      A       = reshape([-0.5],1,1)
              = reshape([1],1,1)
      Xo      = [1]
      t_disc  = 100
      gap     = 1

      # Get full model run
      Random.seed!(2014)
      X = modgen_LSDE(t_start,t_stop,h;
                     A, , Xo, t_disc, gap)
```

```
[2]: 1×90001 Array{Float64,2}:  
      -0.613524 -0.534909 -0.475822 ... -0.823833 -0.786908 -0.842031
```

```
[4]: # Put in Psi functions  
Psi(x) = x
```

```
[4]: Psi (generic function with 1 method)
```

```
[5]: # Model reduction Parameters  
M_out = 100  
n = 2  
p = 500  
par = 55  
ty = "bin"  
rl = true  
Preds = true  
PI = false  
rtol = 1e-6  
  
nfft = 1024  
  
@time h_wf, pred = get_wf(X, Psi;  
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);  
  
X_sig = X[:,2:end]
```

```
Number of CKMS iterations: 133  
errK errR : 2.6568751778621344e-11 1.415333433858973e-14  
10.773960 seconds (35.35 M allocations: 1.790 GiB, 6.02% gc time)
```

```
[5]: 1×90000 Array{Float64,2}:  
      -0.534909 -0.475822 -0.485553 ... -0.823833 -0.786908 -0.842031
```

```
[6]: pred
```

```
[6]: 1×90000 Array{Complex{Float64},2}:  
      -0.613524+0.0im -0.534909+0.0im ... -0.823833+0.0im -0.786908+0.0im
```

1.2 Analyze Wiener Filter

```
[7]: h_wf
```

```
[7]: 1×1×100 Array{Float64,3}:  
[:, :, 1] =  
0.9919349839409488  
  
[:, :, 2] =
```

```
-0.0018356819685355493
```

```
[:, :, 3] =  
0.005150154633584135
```

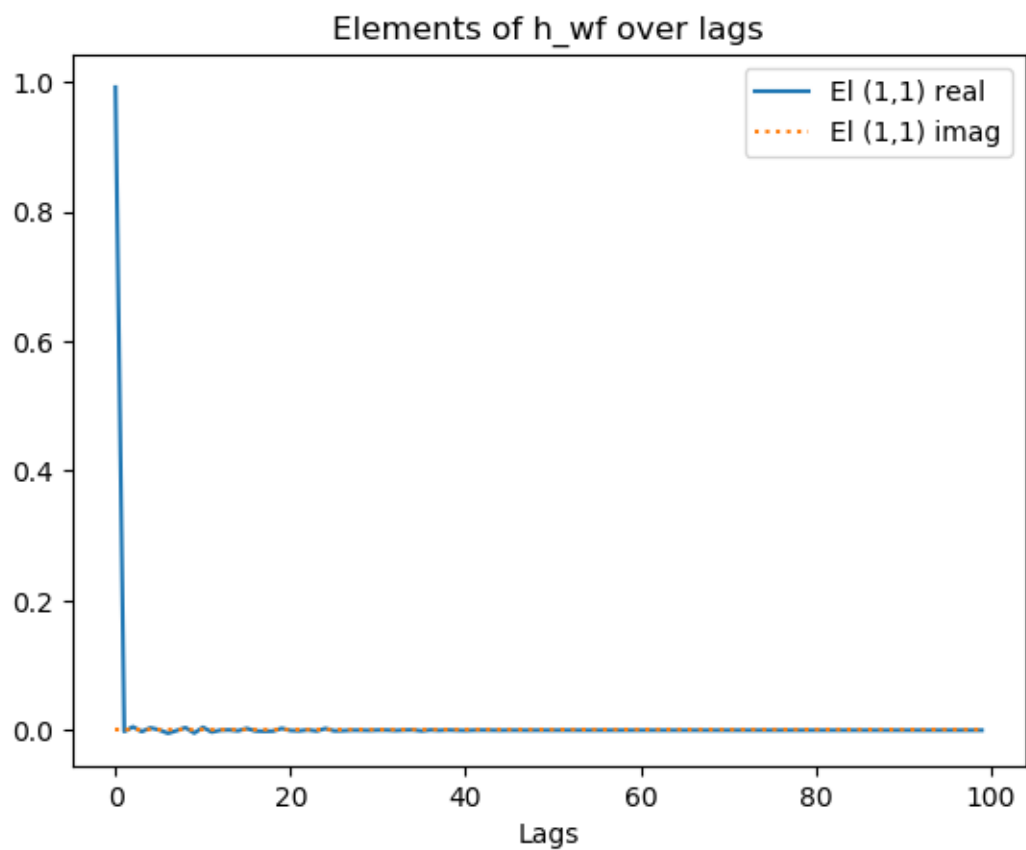
```
...
```

```
[:, :, 98] =  
-3.319811659733008e-7
```

```
[:, :, 99] =  
-1.780465376648296e-7
```

```
[:, :, 100] =  
9.266884051820623e-8
```

```
[8]: plot_WF(h_wf, rl = false)
```



```
[8]: PyObject Text(0.5, 1, 'Elements of h_wf over lags')
```

1.2.1 Poles of Wiener Filter

```
[ ]:
```

```
[ ]:
```

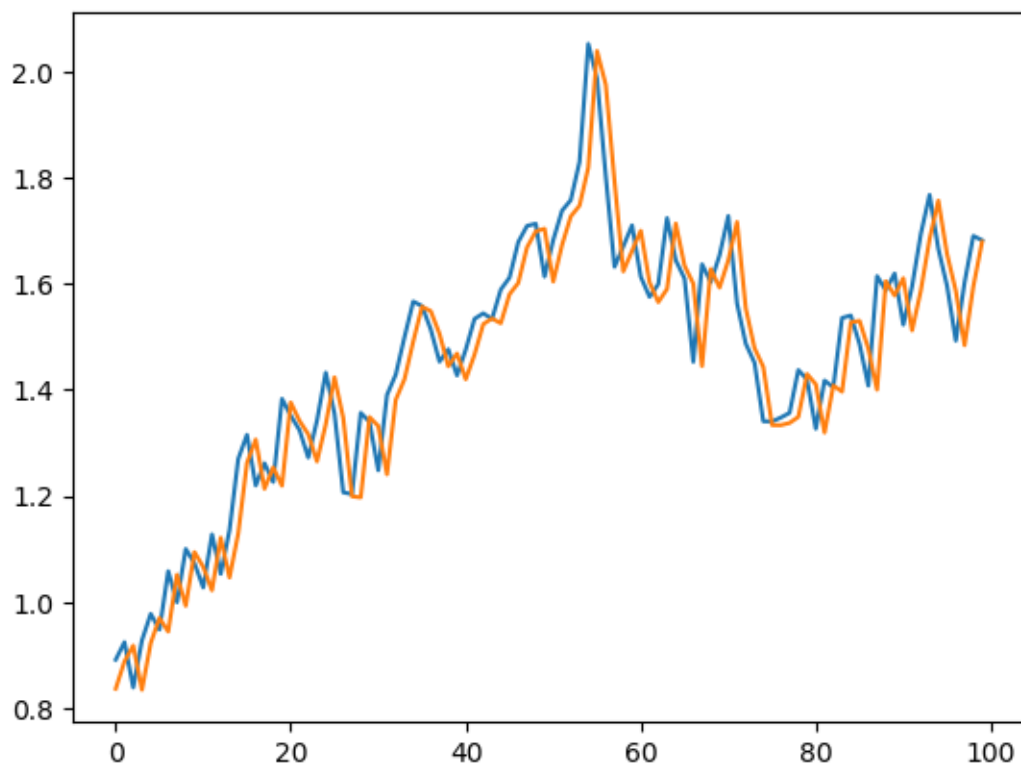
```
[ ]:
```

1.3 Analyze One step prediction error

```
[9]: X_hat = one_step_pred(X_sig, h_wf, pred)
```

```
[9]: 1×90000 Array{Complex{Float64},2}:  
      -0.534909+0.0im -0.475822+0.0im ... -0.817899+0.0im -0.783318+0.0im
```

```
[10]: wind = (1:100) .+ 24000  
      plot([X_sig[1,wind] X_hat[1,wind]])
```



```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:  
ComplexWarning: Casting complex values to real discards the imaginary part  
return array(a, dtype, copy=False, order=order)
```

```
[10]: 2-element Array{PyObject,1}:
      PyObject <matplotlib.lines.Line2D object at 0x0000000061DE78C8>
      PyObject <matplotlib.lines.Line2D object at 0x0000000061C73D48>
```

```
[11]: X_err = X_sig - X_hat

      C = my_crosscov(pred[:,X_err[:],-100:10)

      plot(C)
```

```
[11]: 1×90000 Array{Complex{Float64},2}:
      0.0-0.0im  0.0-0.0im  0.0-0.0im  ...  0.0309909-0.0im  -0.0587131-0.0im
```

1.4 Analyze Reduced Model Run

```
[95]: X_rm = redmodrun(X, h_wf, Psi)
```

reduced model did not blowup

```
[95]: 1×90001 Array{Complex{Float64},2}:
      0.126929+0.0im  0.220936+0.0im  ...  -0.900884+0.0im  0.388956+0.0im
```

```
[ ]:
```

```
[ ]:
```

1.5 2-D Nondiagonal

```
[97]: B = -[-0.5 1; 0 -0.2]*[-0.5 1; 0 -0.2]'/1.5
      B
```

```
[97]: 2×2 Array{Float64,2}:
      -0.833333  0.133333
      0.133333  -0.0266667
```

```
[161]: # Model run Parameters
      t_start = 0
      t_stop  = 1e4
      h       = 1e-2

      A       = B
           = [1 0; 0 1]
      Xo      = [1; 1]
      t_disc  = 100
      gap     = 1

      # Get full model run
```

```
Random.seed!(2017)
Y = modgen_LSDE(t_start,t_stop,h;
    A, , Xo, t_disc, gap)
```

```
[161]: 2×990001 Array{Float64,2}:
 0.98628  0.98338  0.900196  0.971263  ...  -1.33087  -1.40521  -1.44837
 5.7613   5.71792  5.53555   5.58036      -15.1583  -15.1482  -15.1142
```

```
[162]: # Put in Psi functions
Psi(x) = x

# Model reduction Parameters
M_out = 100
n = 2
p = 100
par = 55
ty = "bin"
rl = true
Preds = true
PI = false
rtol = 1e-6

nfft = 1024

@time h_wf_both, pred = get_wf(Y, Psi;
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);

Y_sig = Y[:,2:end]
```

```
Number of CKMS iterations: 491
errK errR : 8.609094825013404e-11 2.1557086806087588e-15
2.381258 seconds (1.13 M allocations: 1.621 GiB, 4.94% gc time)
```

```
[162]: 2×990000 Array{Float64,2}:
 0.98338  0.900196  0.971263  0.994789  ...  -1.33087  -1.40521  -1.44837
 5.71792  5.53555   5.58036   5.41458      -15.1583  -15.1482  -15.1142
```

```
[163]: h_wf_both
```

```
[163]: 2×2×100 Array{Float64,3}:
[:, :, 1] =
 0.992474  0.00109992
 0.00214756 0.999657

[:, :, 2] =
-0.000793588 5.73676e-5
 0.000513344 0.000241174
```

```
[:, :, 3] =
  0.000869982 -0.000286125
 -0.00122289 -0.00119846
```

...

```
[:, :, 98] =
 -2.76064e-6 -1.45713e-5
 -3.63079e-6 -2.02967e-5
```

```
[:, :, 99] =
  1.38051e-6  1.0264e-5
 -3.60786e-6 -1.98643e-5
```

```
[:, :, 100] =
 -3.76341e-7 -3.67231e-7
  9.09628e-7  7.93528e-6
```

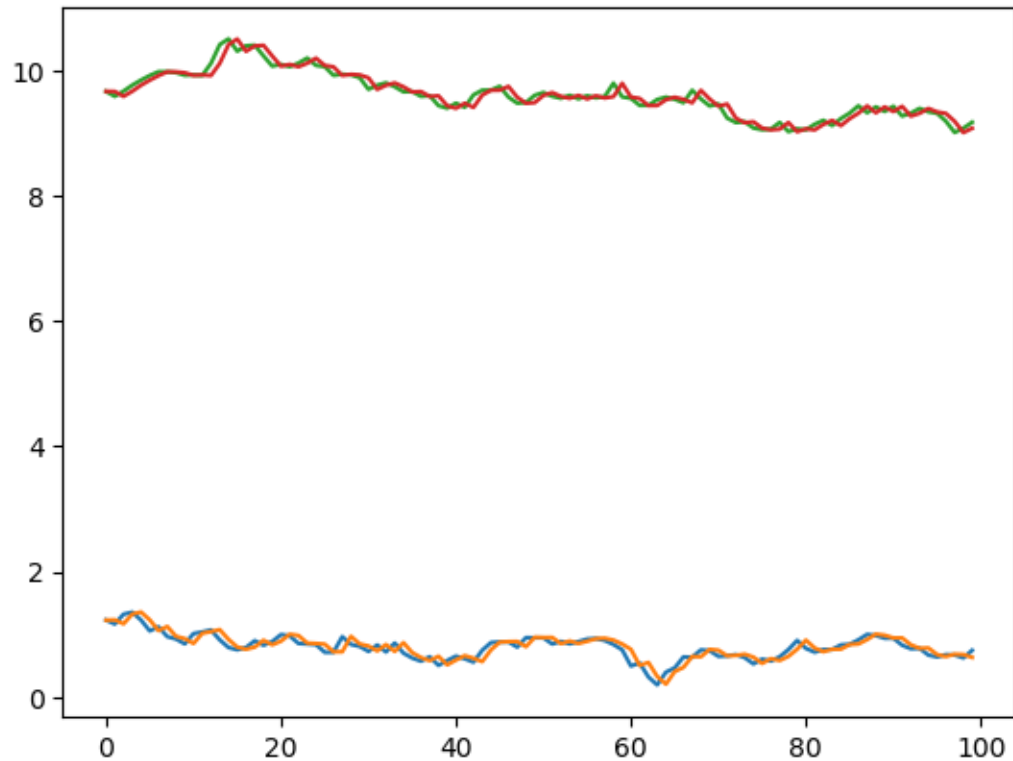
```
[164]: I+h*B
```

```
[164]: 2x2 Array{Float64,2}:
  0.991667  0.00133333
  0.00133333  0.999733
```

```
[165]: Y_hat = one_step_pred(Y_sig, h_wf_both, pred)
```

```
[165]: 2x990000 Array{Complex{Float64},2}:
  0.98338+0.0im  0.900196+0.0im  ...  -1.34007+0.0im  -1.4145+0.0im
  5.71792+0.0im  5.53555+0.0im      -15.1553+0.0im  -15.146+0.0im
```

```
[166]: wind = (1:100) .+ 24000
plot([Y_sig[1,wind] Y_hat[1,wind] Y_sig[2,wind] Y_hat[2,wind]])
```

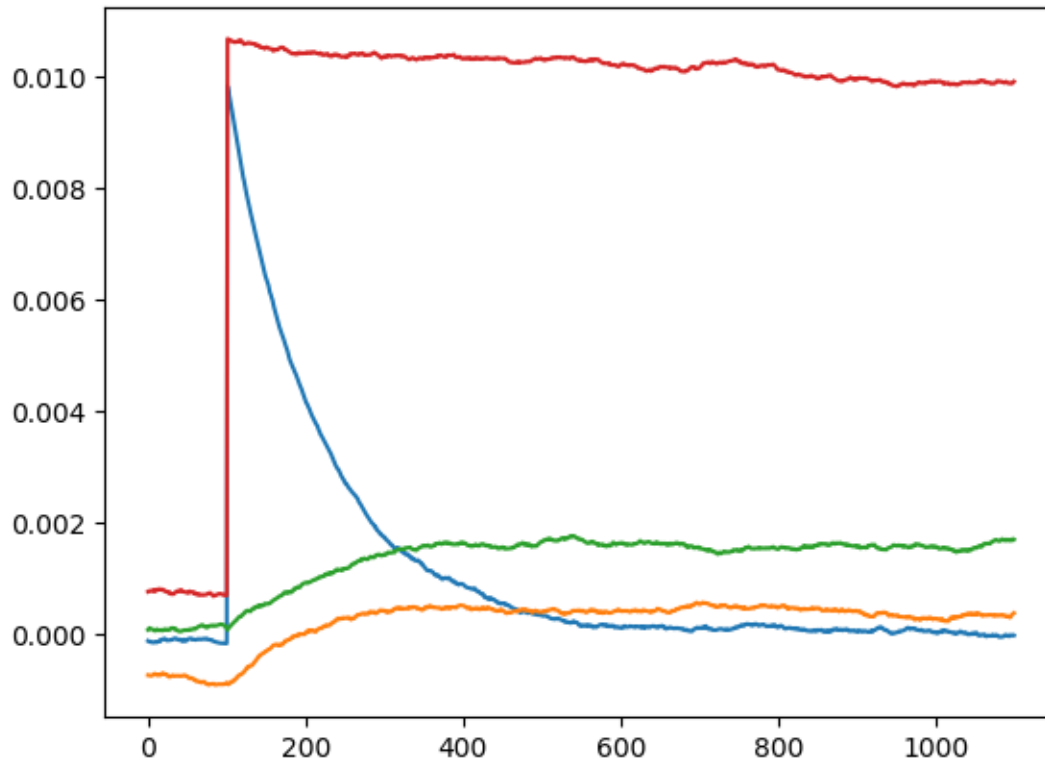


```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[166]: 4-element Array{PyObject,1}:
 PyObject <matplotlib.lines.Line2D object at 0x00000000E54391C8>
 PyObject <matplotlib.lines.Line2D object at 0x00000000E58ACC48>
 PyObject <matplotlib.lines.Line2D object at 0x00000000E58AC0C8>
 PyObject <matplotlib.lines.Line2D object at 0x00000000E58AC3C8>
```

```
[167]: Y_err = Y_sig - Y_hat
lags = -100:1000
C11 = my_crosscov(pred[1,:],Y_err[1,:],lags)
C21 = my_crosscov(pred[2,:],Y_err[1,:],lags)
C12 = my_crosscov(pred[1,:],Y_err[2,:],lags)
C22 = my_crosscov(pred[2,:],Y_err[2,:],lags)

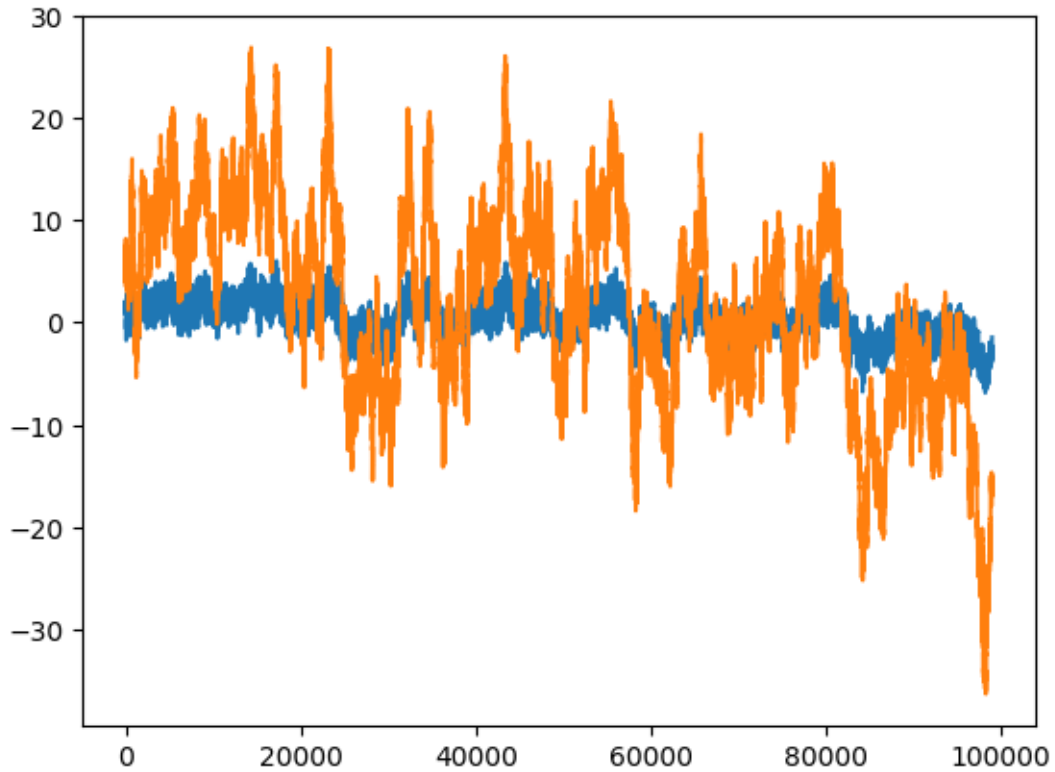
plot([C11 C21 C12 C22])
```

```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[167]: 4-element Array{PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x00000000E580B308>
PyObject <matplotlib.lines.Line2D object at 0x00000000E58164C8>
PyObject <matplotlib.lines.Line2D object at 0x00000000E58050C8>
PyObject <matplotlib.lines.Line2D object at 0x00000000E5805E48>
```

```
[168]: plot(Y[:,1:10:end]')
```



```
[168]: 2-element Array{PyCall.PyObject,1}:
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8119148>
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8120CC8>
```

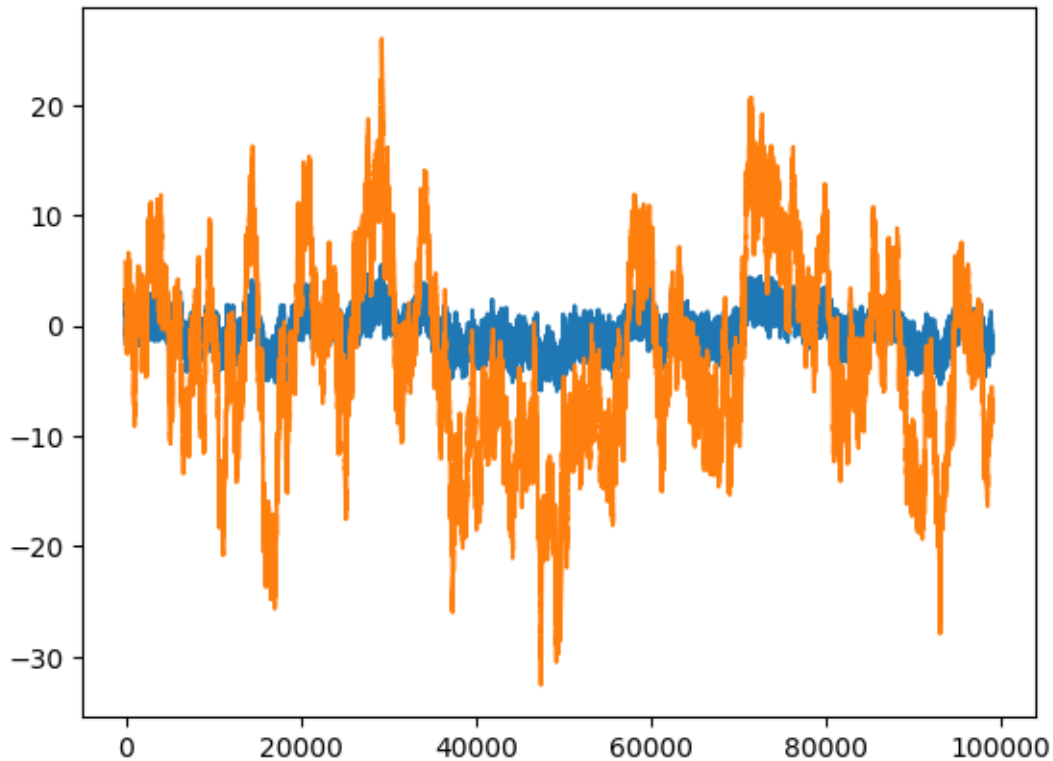
```
[169]: noise_dist = MvNormal(h*I + zeros(size(Y,1),size(Y,1)))

        Y_rm = redmodrun(Y, h_wf, Psi;noise_dist)
```

reduced model did not blowup

```
[169]: 2×990001 Array{Complex{Float64},2}:
 0.98628+0.0im  0.98338+0.0im  ...  -2.00373+0.0im  -2.22789+0.0im
 5.7613+0.0im  5.71792+0.0im      -8.41249+0.0im  -8.44843+0.0im
```

```
[170]: plot(Y_rm[:,1:10:end]')
```



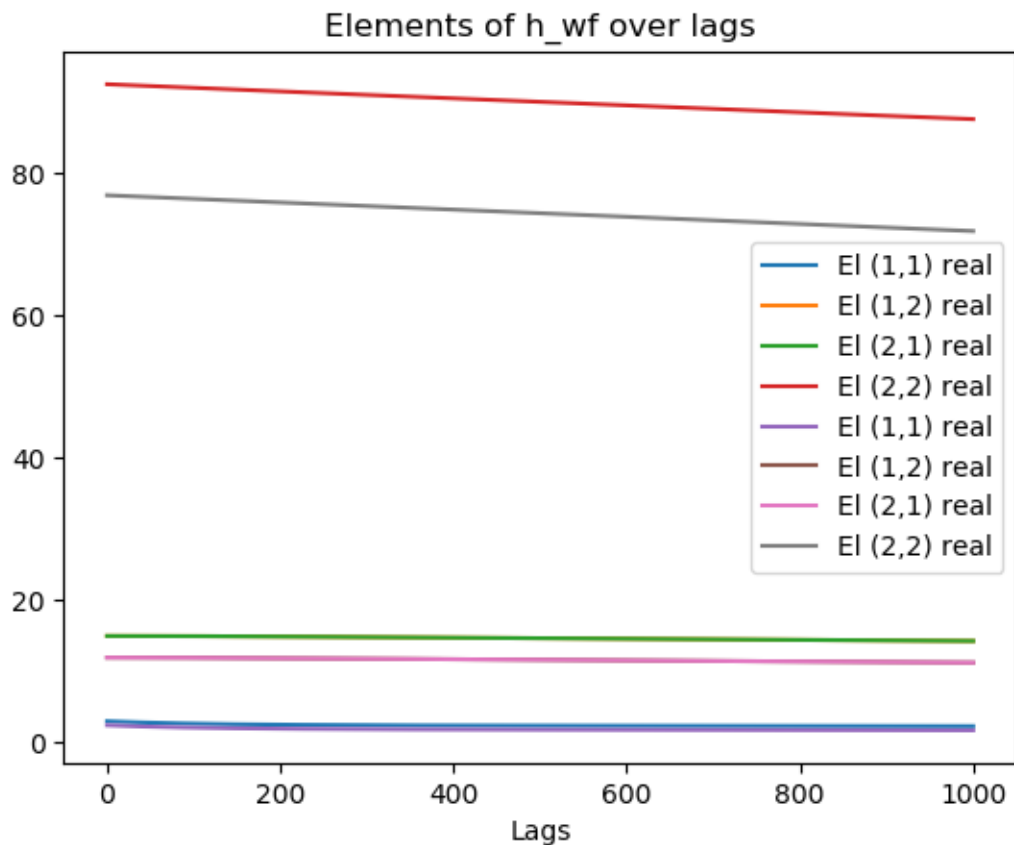
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)

```
[170]: 2-element Array{PyObject,1}:
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8180088>
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8186C08>
```

```
[171]: C = matrix_autocov_seq(Y;
        L = 1000, win = "par"
        )

        C_rm = matrix_autocov_seq(Y_rm;
        L = 1000, win = "par"
        )

        plot_WF(C)
        plot_WF(C_rm)
```



[171]: PyObject Text(0.5, 1, 'Elements of h_wf over lags')

1.6 Model reduction

```
[172]: # Put in Psi functions
Psi(x) = x

# Model reduction Parameters
M_out = 500
n = 2
p = 500
par = 55
ty = "bin"
rl = true
Preds = true
PI = false
rtol = 1e-6

nfft = 1024
```

```
@time h_wf_1, pred = get_wf(Y[1:1,:], Psi;
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);

Y_sig_1 = Y[1:1,2:end]
```

```
Number of CKMS iterations: 172
errK errR : 3.676771965803039e-11 1.208942325415465e-15
0.594519 seconds (1.10 M allocations: 506.237 MiB, 4.06% gc time)
```

```
[172]: 1×990000 Array{Float64,2}:
0.98338 0.900196 0.971263 0.994789 ... -1.33087 -1.40521 -1.44837
```

```
[173]: h_wf_1
```

```
[173]: 1×1×500 Array{Float64,3}:
[:, :, 1] =
0.9984574296698595
```

```
[:, :, 2] =
-0.0007074519521541362
```

```
[:, :, 3] =
0.000985466803211898
```

```
...
```

```
[:, :, 498] =
-3.8979099752576554e-11
```

```
[:, :, 499] =
-3.436200769119326e-11
```

```
[:, :, 500] =
-2.7820155495252038e-11
```

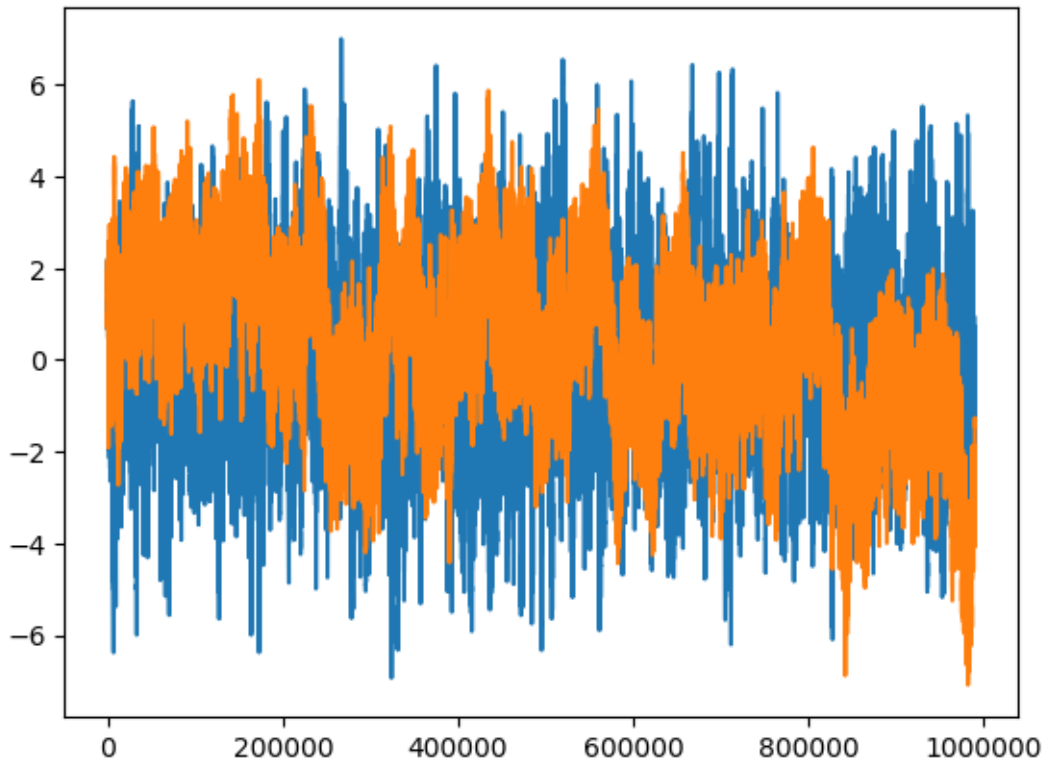
```
[174]: noise_dist = MvNormal(h*I + zeros(1,1))

Y_rm_1 = redmodrun(real(Y[1:1,:]), h_wf_1, Psi;noise_dist)
```

```
reduced model did not blowup
```

```
[174]: 1×990001 Array{Complex{Float64},2}:
0.98628+0.0im 0.98338+0.0im ... -0.103294+0.0im -0.269171+0.0im
```

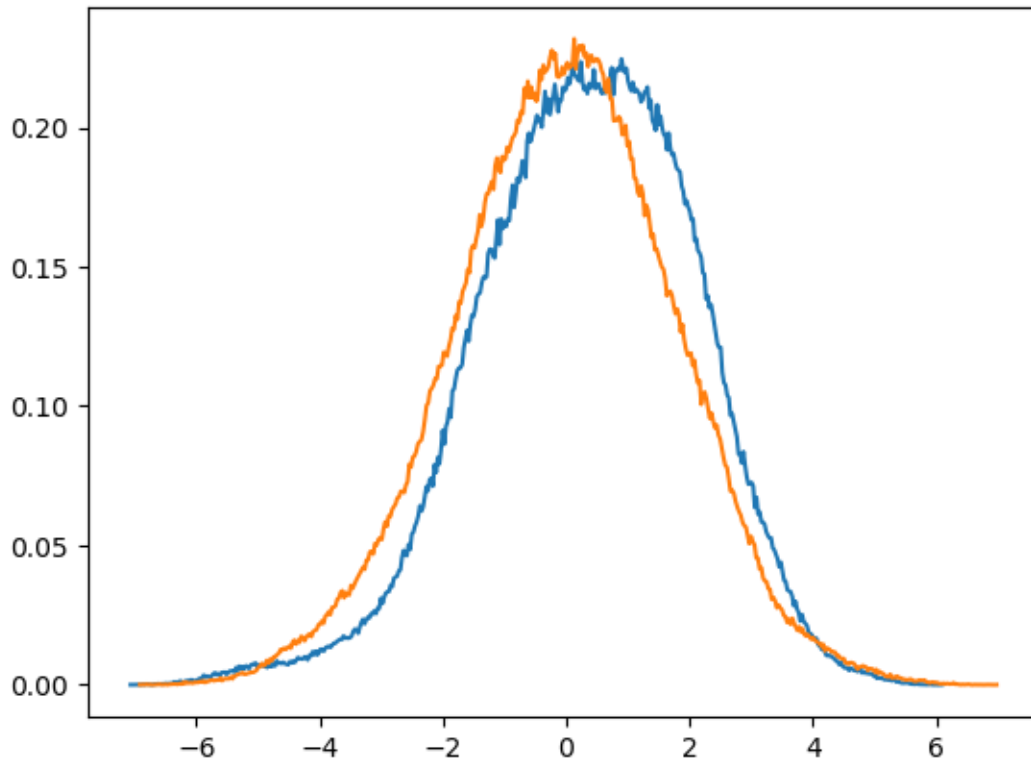
```
[175]: plot([Y_rm_1[:, :] Y[1,:]])
```



```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[175]: 2-element Array{PyCall.PyObject,1}:
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8193348>
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8198EC8>
```

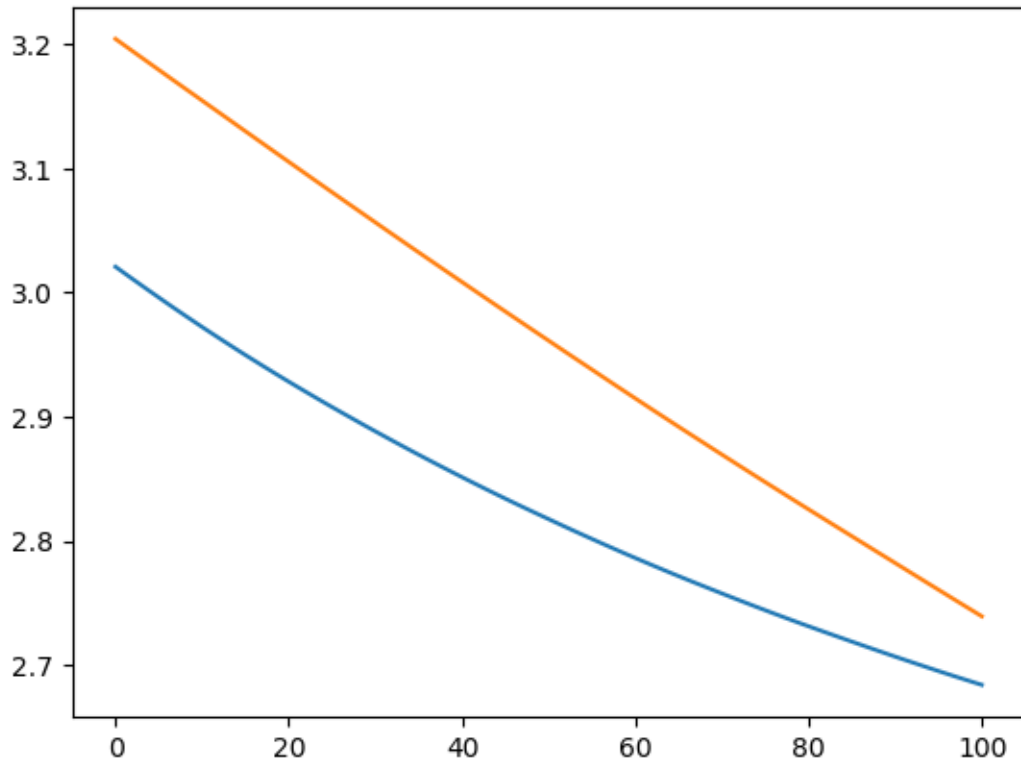
```
[176]: emp_pdf(real(Y[1,:]),bn = 500)
        emp_pdf(real(Y_rm_1[:]),bn = 500)
```



```
[176]: 3-element Array{Array{T,1} where T,1}:
 [7.25229397951427e-5, 0.0, 0.00014504587959028543, 3.6261469897571324e-5,
 0.0002175688193854281, 7.252293979514272e-5, 0.00014504587959028538, 0.0,
 3.626146989757142e-5, 0.00014504587959028543 ... 7.252293979655886e-5, 0.0,
 3.626146989628664e-5, 7.252293979655886e-5, 7.252293979655886e-5,
 3.626146989628664e-5, 0.0, 3.626146989628664e-5, 0.0, 3.6261469900272214e-5]
 [-6.9178494106116855, -6.889993397255051, -6.862137383898416,
 -6.834281370541779, -6.8064253571851445, -6.778569343828509, -6.750713330471873,
 -6.722857317115238, -6.695001303758603, -6.667145290401967 ...
 6.731597134139671, 6.759453147496306, 6.787309160852941, 6.815165174209577,
 6.843021187566213, 6.870877200922847, 6.898733214279483, 6.926589227636119,
 6.9544452409927535, 6.982301254349389]
 PyCall.PyObject[PyObject <matplotlib.lines.Line2D object at
 0x00000000F2D2BB08>]
```

```
[177]: lags = 0:100
A_1 = my_autocov(real(Y[1,:]),lags)
A_rm_1 = my_autocov(real(Y_rm_1[:]),lags)

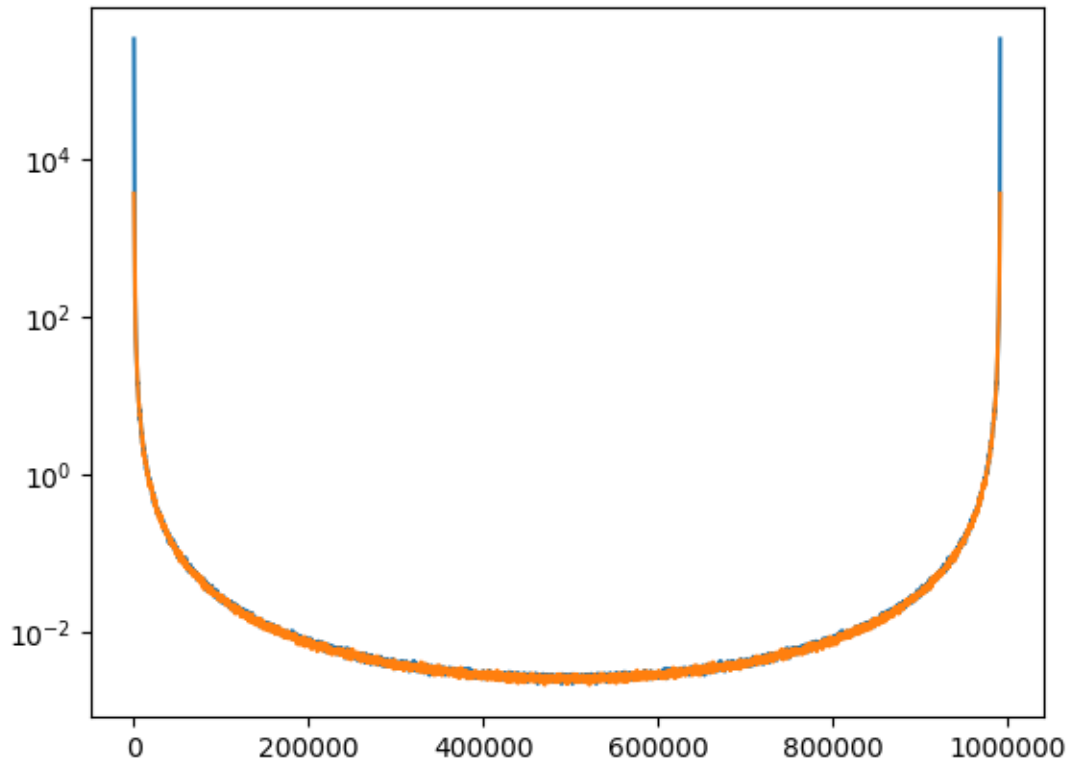
plot([A_1 A_rm_1])
```



```
[177]: 2-element Array{PyCall.PyObject,1}:
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8569288>
        PyObject <matplotlib.lines.Line2D object at 0x00000000E8570E08>
```

```
[178]: S_1 = z_crossspect_scalar(real(Y[2,:]),real(Y[2,:]))
        S_rm_1 = z_crossspect_scalar(real(Y_rm_1[1,:]),real(Y_rm_1[1,:]))

        semilogy([S_1 S_rm_1])
```

C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)

[178]: 2-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x00000000E85A7CC8>
PyObject <matplotlib.lines.Line2D object at 0x00000000E85D9808>

```
[179]: # Put in Psi functions
Psi(x) = x

# Model reduction Parameters
M_out = 100
n = 2
p = 500
par = 55
ty = "bin"
rl = true
Preds = true
PI = false
rtol = 1e-6
```

```

nfft = 1024

@time h_wf_2, pred = get_wf(Y[2:2,:], Psi;
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);

Y_sig_2 = Y[:,2:end]

```

```

Number of CKMS iterations: 590
errK errR : 2.047749177425513e-11 1.3745917404177208e-15
0.567783 seconds (1.12 M allocations: 515.725 MiB, 3.09% gc time)

```

```

[179]: 2×990000 Array{Float64,2}:
 0.98338  0.900196  0.971263  0.994789  ...  -1.33087  -1.40521  -1.44837
 5.71792  5.53555   5.58036  5.41458   ...  -15.1583  -15.1482  -15.1142

```

```

[180]: h_wf_2

```

```

[180]: 1×1×100 Array{Float64,3}:
[:, :, 1] =
 1.0000061932826605

[:, :, 2] =
 0.00026021522262930806

[:, :, 3] =
 -0.0012670873060590824

...

[:, :, 98] =
 -1.9326221730464675e-5

[:, :, 99] =
 -1.9429717047415154e-5

[:, :, 100] =
 6.8522680176718665e-6

```

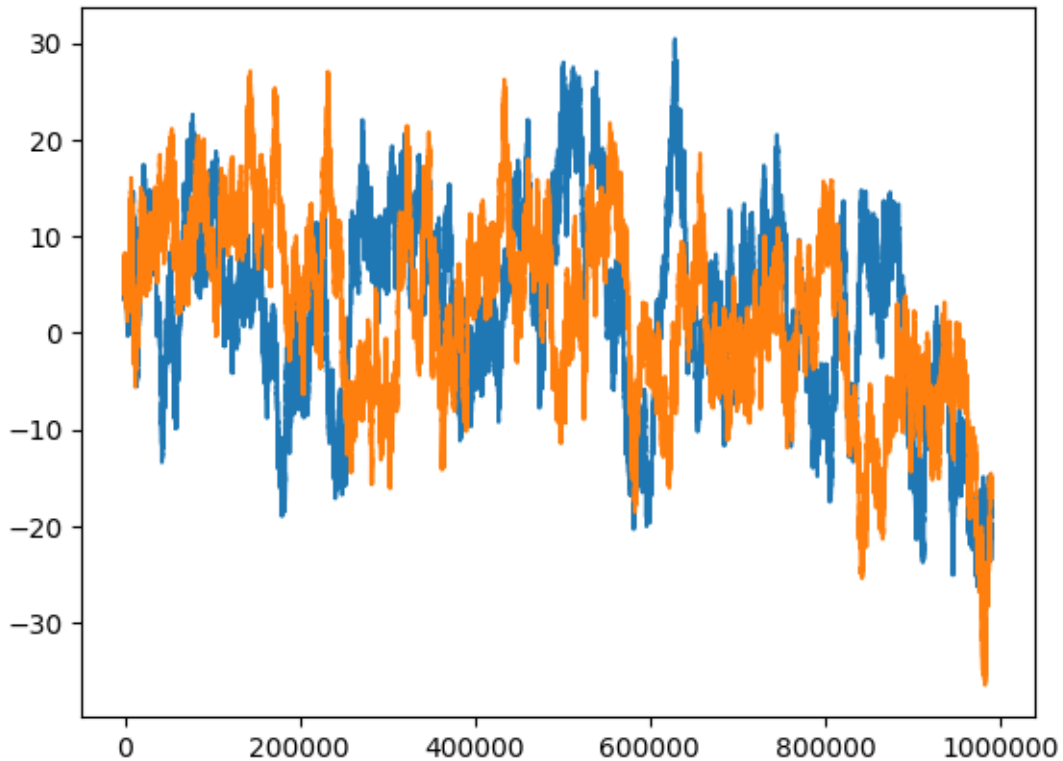
```

[181]: noise_dist = MvNormal(h*I + zeros(1,1))

Y_rm_2 = redmodrun(real(Y[2:2,:]), h_wf_2, Psi;noise_dist)

plot([Y_rm_2[:, Y[2,:]])

```

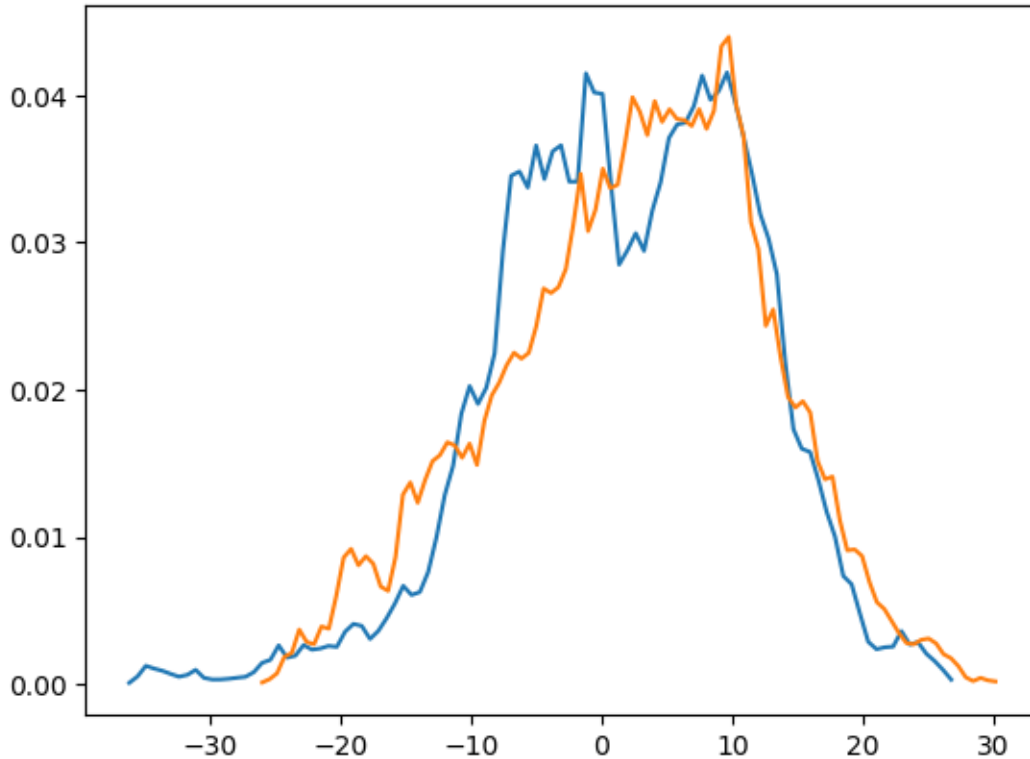


reduced model did not blowup

```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[181]: 2-element Array{PyCall.PyObject,1}:
        PyObject <matplotlib.lines.Line2D object at 0x00000000F9260848>
        PyObject <matplotlib.lines.Line2D object at 0x00000000F9279BC8>
```

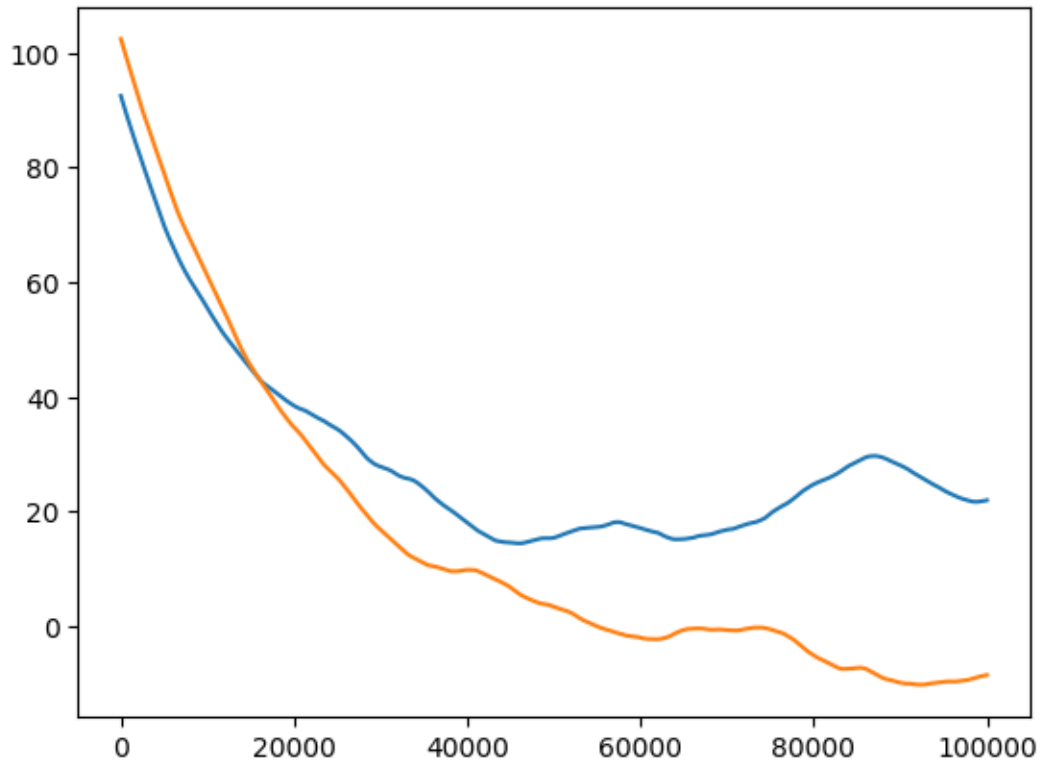
```
[182]: emp_pdf(real(Y[2,:]),bn = 100)
        emp_pdf(real(Y_rm_2[:]),bn = 100)
```



```
[182]: 3-element Array{Array{T,1} where T,1}:
 [0.00010522989253827487, 0.00032995813762001444, 0.0007241243452633831,
 0.0018566833581753243, 0.002111732080768092, 0.0036955311413441622,
 0.0028590426735399087, 0.0027003060559821707, 0.003907774483921361,
 0.003759739211367517 ... 0.0030873380336229514, 0.0027555963385022795,
 0.002027904878237504, 0.0017550205806383112, 0.0012181697729432453,
 0.0004583742776665578, 0.00019975844007283342, 0.0004155688976511327,
 0.0002479144925900151, 0.0001747886350636557]
 [-25.948020554845122, -25.38168053528837, -24.81534051573162,
 -24.249000496174865, -23.682660476618118, -23.116320457061363,
 -22.549980437504612, -21.98364041794786, -21.417300398391106,
 -20.850960378834355 ... 25.02258120526256, 25.588921224819316,
 26.155261244376067, 26.72160126393282, 27.28794128348957, 27.85428130304632,
 28.420621322603075, 28.986961342159827, 29.553301361716578, 30.11964138127333]
 PyCall.PyObject[PyObject <matplotlib.lines.Line2D object at
 0x00000000E8630988>]
```

```
[183]: lags = 0:100000
A_2 = my_autocov(real(Y[2,:]),lags)
A_rm_2 = my_autocov(real(Y_rm_2[:]),lags)

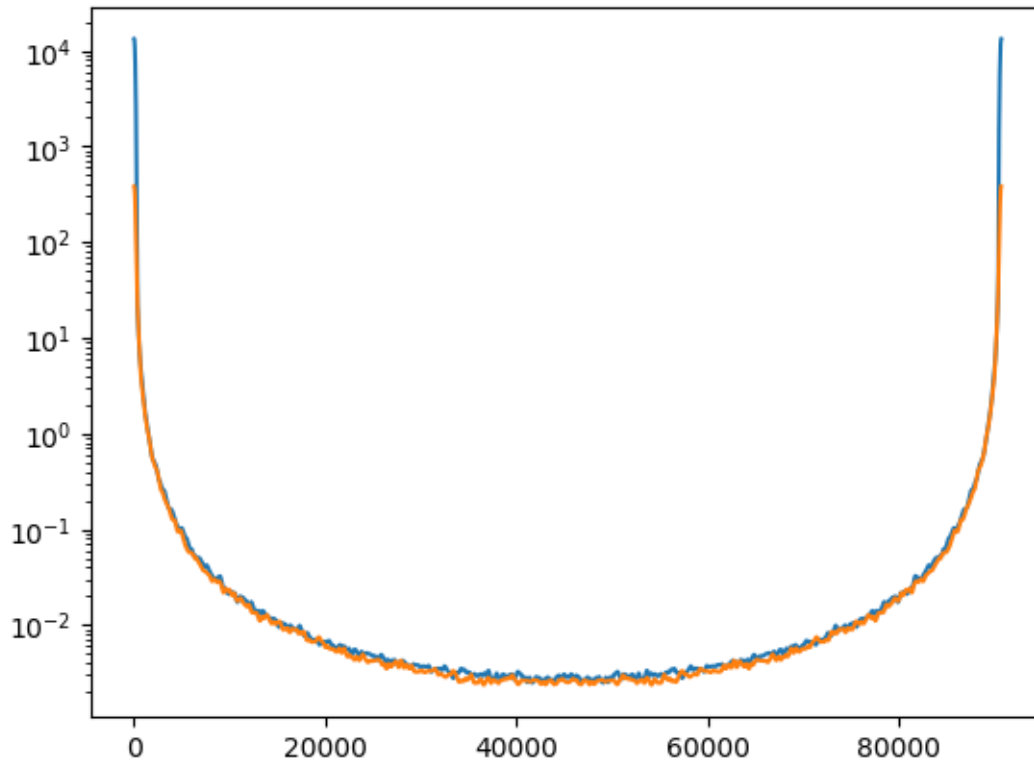
plot([A_2 A_rm_2])
```



```
[183]: 2-element Array{PyCall.PyObject,1}:
        PyObject <matplotlib.lines.Line2D object at 0x00000000FC40AE48>
        PyObject <matplotlib.lines.Line2D object at 0x00000000FC411308>
```

```
[184]: S_2 = z_crossspect_scalar(real(Y[2,:]),real(Y[2,:]))
        S_rm_2 = z_crossspect_scalar(real(Y_rm_2[1,:]),real(Y_rm_2[1,:]))

        semilogy([S S_rm])
```



```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[184]: 2-element Array{PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x00000000FC43D4C8>
PyObject <matplotlib.lines.Line2D object at 0x00000000FBA12D48>
```

```
[ ]:
```

1.7 7-D Nondiagonal

```
[16]: d = 7

b = randn(d,d)
B = -b*b'/(svd(b*b').S[1] + 1)
```

```
[16]: 7×7 Array{Float64,2}:
-0.429329  -0.0841054   0.0557147   ...   0.121911  -0.0168159  -0.236922
-0.0841054  -0.482409  -0.262653   ...   0.237132  -0.037892  -0.00570117
 0.0557147  -0.262653  -0.316618   ...   0.114749  -0.131388   0.111828
 0.0732418   0.151468   0.0297526   ...  -0.198649  -0.0999181   0.07232
```

```

0.121911    0.237132    0.114749    -0.415005   -0.133163   -0.0506283
-0.0168159 -0.037892    -0.131388    ...   -0.133163   -0.401556    0.0476175
-0.236922   -0.00570117   0.111828    -0.0506283   0.0476175   -0.268553

```

```
[17]: isposdef(-B)
```

```
[17]: true
```

```
[18]: # Model run Parameters
t_start = 0
t_stop  = 1e4
h       = 1e-2

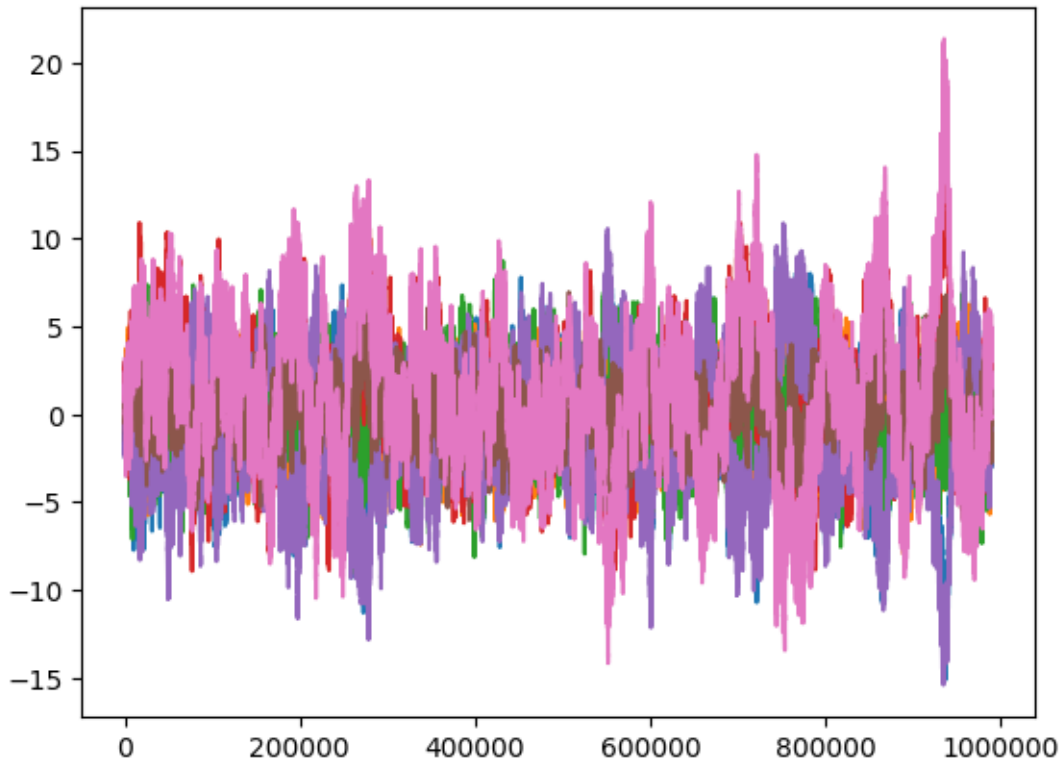
A       = B
        = I + zeros(d,d)
Xo      = ones(d)
t_disc  = 100
gap     = 1

# Get full model run
Random.seed!(2017)
Y = modgen_LSDE(t_start,t_stop,h;
    A, , Xo, t_disc, gap)
```

```
[18]: 7×990001 Array{Float64,2}:
-0.22646   -0.182972  -0.218403  ...  -0.920501  -0.987486  -0.953155
-0.162249  -0.201968  -0.0401256 ...  -0.634519  -0.670849  -0.655005
-0.0760985 -0.107251  -0.190555   1.17574   1.01653    1.0076
 2.62242   2.58614   2.35267    3.1203    3.24898    3.20971
-0.349861  -0.488975  -0.513428  -0.679365  -0.64448   -0.669547
-0.239206  -0.373286  -0.434784  ...  -2.51162   -2.39705   -2.40973
 3.13468    3.10144    3.03561   -0.167722  -0.270254  -0.190301

```

```
[20]: plot(Y')
```



```
[20]: 7-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x0000000053A44D48>
PyObject <matplotlib.lines.Line2D object at 0x0000000053A57188>
PyObject <matplotlib.lines.Line2D object at 0x0000000053A57348>
PyObject <matplotlib.lines.Line2D object at 0x0000000053A57508>
PyObject <matplotlib.lines.Line2D object at 0x0000000053A57788>
PyObject <matplotlib.lines.Line2D object at 0x0000000053A57A08>
PyObject <matplotlib.lines.Line2D object at 0x0000000053A57C08>
```

```
[24]: # Put in Psi functions
Psi(x) = x

# Model reduction Parameters
M_out = 100
n = 2
p = 100
par = 500
ty = "bin"
rl = true
Preds = true
PI = false
rtol = 1e-6
```



```

nfft = 2^16

@time h_wf_all, Y_pred = get_wf(Y[:, :], Psi;
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);

Y_sig = Y[:, 2:end]

```

Number of CKMS iterations: 1922

errK errR : 8.199708649445936e-11 9.871650104093513e-16

47.301215 seconds (5.53 M allocations: 26.154 GiB, 3.41% gc time)

[24]: 7×990000 Array{Float64,2}:

```

-0.182972 -0.218403 -0.203021 ... -0.920501 -0.987486 -0.953155
-0.201968 -0.0401256 0.0412264 -0.634519 -0.670849 -0.655005
-0.107251 -0.190555 -0.115105 1.17574 1.01653 1.0076
2.58614 2.35267 2.41436 3.1203 3.24898 3.20971
-0.488975 -0.513428 -0.455456 -0.679365 -0.64448 -0.669547
-0.373286 -0.434784 -0.417762 ... -2.51162 -2.39705 -2.40973
3.10144 3.03561 2.95586 -0.167722 -0.270254 -0.190301

```

[26]: h_wf_all

[26]: 7×7×100 Array{Float64,3}:

[:, :, 1] =

```

0.993441 0.000864392 1.87734e-5 ... 0.0100773 -0.00900526
0.00129688 0.997155 -0.00369679 0.00387807 0.00639146
0.00420231 -0.0002035 0.993934 -0.0111595 0.00852533
0.00201841 0.00311996 -0.00421354 0.00358555 -0.00355979
-0.00805861 0.00759582 -0.00267206 0.00520624 -0.00537361
-0.00830386 -0.00448481 0.00747978 ... 0.995458 0.000954408
0.00676175 -0.0136123 -0.00739882 -0.00403972 1.00087

```

[:, :, 2] =

```

0.00101765 0.000138468 -0.000955201 ... -0.000163965 3.38761e-5
0.000915134 -0.00163753 -0.000550461 0.000867228 -0.0004428
-0.00206755 6.29378e-5 -0.000517688 -0.000380712 -0.000167938
-0.000975924 0.000650634 -0.000482622 -0.000103971 -0.000973834
-0.000512161 -0.00231568 0.000296131 -0.00188157 -0.00263541
-0.000373563 0.000614138 -0.00178262 ... 0.000344659 -0.00166783
-0.00146959 0.000586171 -0.0001315 -0.00129151 -0.000290608

```

[:, :, 3] =

```

-0.000785355 -0.000438401 0.00141443 ... 0.000599309 0.00132317
0.000477416 -2.22081e-5 -0.00148899 0.00151858 0.000372948
-0.0001765 0.000709857 0.00153736 0.00129883 0.00141307
0.000423839 -0.000225325 -0.00141735 0.00127124 0.00121864

```

```

0.00114507    0.0019372    -0.000714355    0.000690556    0.00324969
5.36609e-5    0.000848252    0.000175294    ... -0.000871651    0.000821109
0.000400672    0.000818503    0.00131045    0.000892226    -0.000777336

...

[:, :, 98] =
-0.000687809 -0.000582107 -0.000694061 ... -0.00111004    0.00202234
0.00108962 -0.00204021 -0.00063664    0.000603063 -0.000179208
0.000817546 0.000812678 -7.81706e-5    -0.00187053    0.000693333
-0.000687759 -0.000351482 -0.000448557    0.0010048    -0.000664821
-0.00095861 -0.000342729 -0.000247914    -0.00142334    0.000891204
0.000455518 -8.36271e-5    0.00105597    ... -0.00147836    -0.000753883
0.000869042 0.000288666 -0.000193672    0.000114234    -0.00189221

[:, :, 99] =
-1.99038e-5    0.000276379    0.000460062 ... 0.00203602    -0.00120803
-0.000250019 0.00232605    0.000874842    -0.000468163    0.000401817
-0.000799394 0.00137212    0.000548446    0.000276954    0.000781679
0.000814795 3.15848e-5    0.000475462    -0.000702103    0.00206405
-0.000345095 -0.00104578    0.000402997    0.000689151    -0.00136999
0.00128205 -0.00106936    -0.000165921 ... 0.000504721    0.000233086
-0.00087294 -0.000205862    0.00114487    -0.000662095    0.000555703

[:, :, 100] =
0.00163823    0.001636    -0.00134187 ... -0.00120059    0.000114441
6.03643e-5    0.00018435    -0.0018989    0.000538329    0.000816033
0.00184235    -0.00212958    -6.14449e-5    0.000410201    -0.00166608
-0.00114493 -8.70622e-5    9.83936e-5    -0.000381936    -0.000442926
0.000641094 -7.22323e-6    -0.000873871    -0.00085601    0.00196516
-0.000375892 4.56433e-5    0.000991723 ... -0.00118287    0.000902092
0.000598694 0.00107411    0.000184504    0.00043699    0.00243592

```

```
[25]: I+h*B
```

```
[25]: 7x7 Array{Float64,2}:
```

```

0.995707    -0.000841054    0.000557147 ... -0.000168159    -0.00236922
-0.000841054 0.995176    -0.00262653    -0.00037892    -5.70117e-5
0.000557147 -0.00262653    0.996834    -0.00131388    0.00111828
0.000732418 0.00151468    0.000297526    -0.000999181    0.0007232
0.00121911 0.00237132    0.00114749    -0.00133163    -0.000506283
-0.000168159 -0.00037892    -0.00131388 ... 0.995984    0.000476175
-0.00236922 -5.70117e-5    0.00111828    0.000476175    0.997314

```

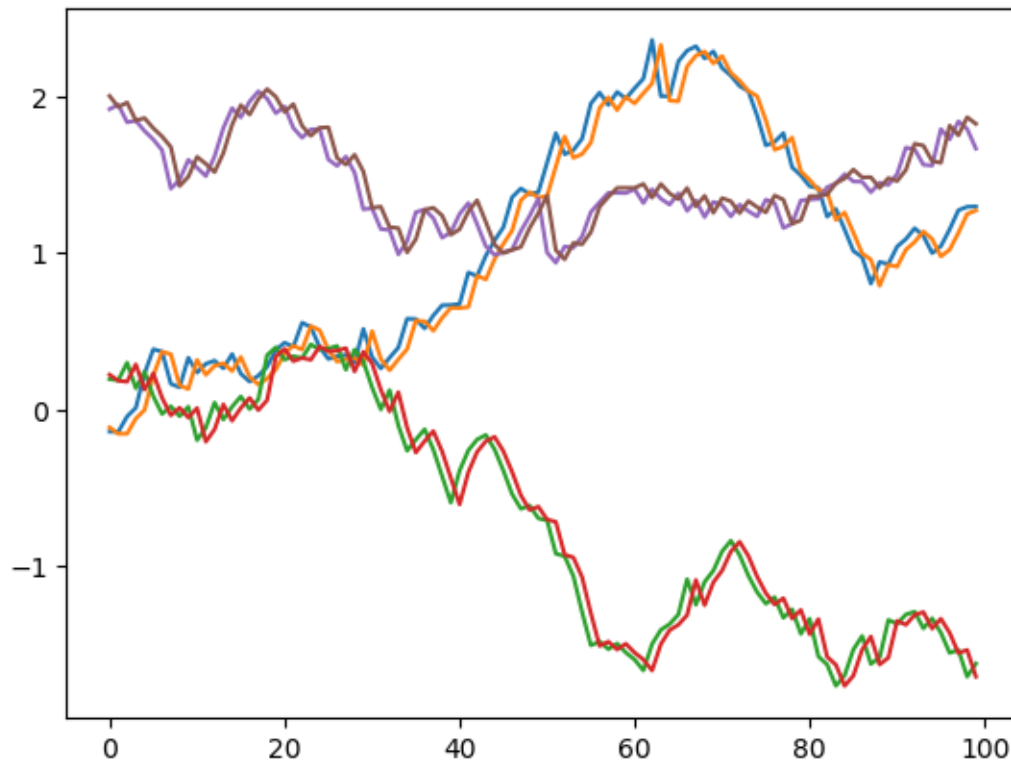
```
[28]: norm(I+h*B - h_wf_all[:, :, 1])
```

```
[28]: 0.04193310984431987
```

```
[29]: Y_hat = one_step_pred(Y_sig, h_wf_all[:, :, 1:10], Y_pred)
```

```
[29]: 7×990000 Array{Complex{Float64},2}:
-0.182972+0.0im -0.218403+0.0im ... -0.943013+0.0im -1.00773+0.0im
-0.201968+0.0im -0.0401256+0.0im -0.667347+0.0im -0.704095+0.0im
-0.107251+0.0im -0.190555+0.0im 1.19983+0.0im 1.04109+0.0im
 2.58614+0.0im 2.35267+0.0im 3.10301+0.0im 3.23289+0.0im
-0.488975+0.0im -0.513428+0.0im -0.69121+0.0im -0.655659+0.0im
-0.373286+0.0im -0.434784+0.0im ... -2.50939+0.0im -2.39628+0.0im
 3.10144+0.0im 3.03561+0.0im -0.134108+0.0im -0.23436+0.0im
```

```
[31]: wind = (1:100) .+ 24000
plot([Y_sig[1,wind] Y_hat[1,wind] Y_sig[2,wind] Y_hat[2,wind] Y_sig[7,wind]
      ↪ Y_hat[7,wind]])
```



```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[31]: 6-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x00000000025A3408>
PyObject <matplotlib.lines.Line2D object at 0x00000000025A9F88>
```

```

PyObject <matplotlib.lines.Line2D object at 0x00000000025B1188>
PyObject <matplotlib.lines.Line2D object at 0x00000000025B1348>
PyObject <matplotlib.lines.Line2D object at 0x00000000025B1548>
PyObject <matplotlib.lines.Line2D object at 0x00000000025B1788>

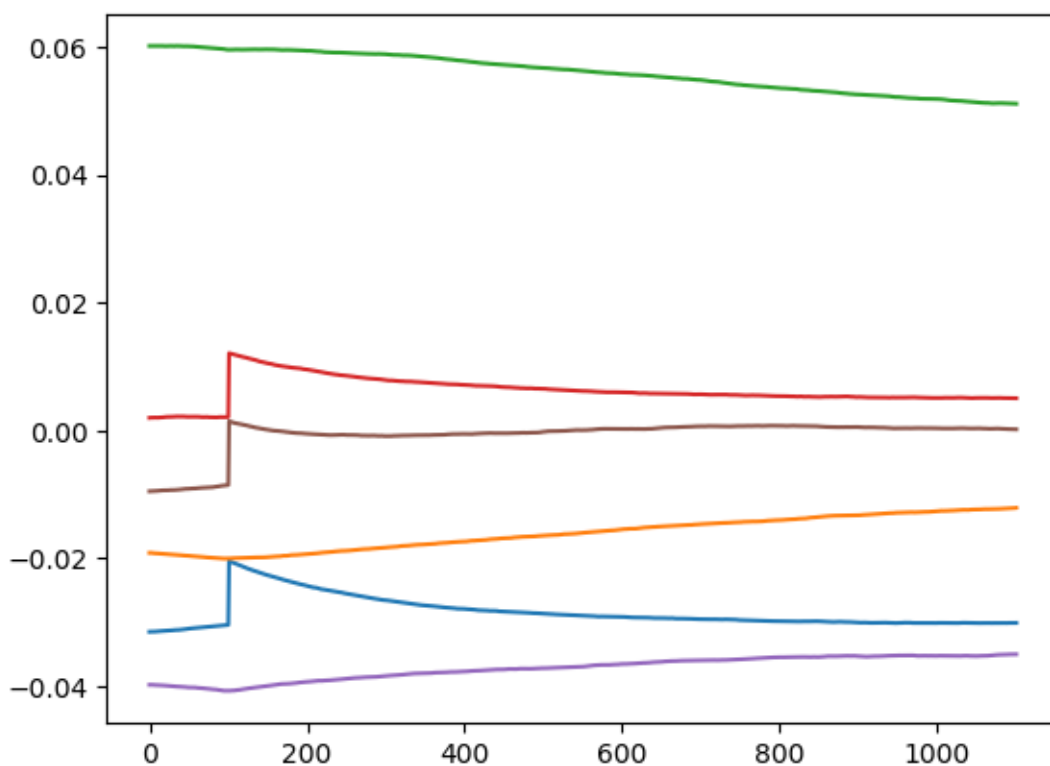
```

```

[35]: Y_err = Y_sig - Y_hat
lags = -100:1000
C11 = my_crosscov(Y_pred[1,:],Y_err[1,:],lags)
C21 = my_crosscov(Y_pred[2,:],Y_err[1,:],lags)
C12 = my_crosscov(Y_pred[1,:],Y_err[2,:],lags)
C22 = my_crosscov(Y_pred[2,:],Y_err[2,:],lags)
C13 = my_crosscov(Y_pred[1,:],Y_err[3,:],lags)
C33 = my_crosscov(Y_pred[3,:],Y_err[3,:],lags)

plot([C11 C21 C12 C22 C13 C33])

```



```

C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)

```

```

[35]: 6-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x000000000548ECD48>

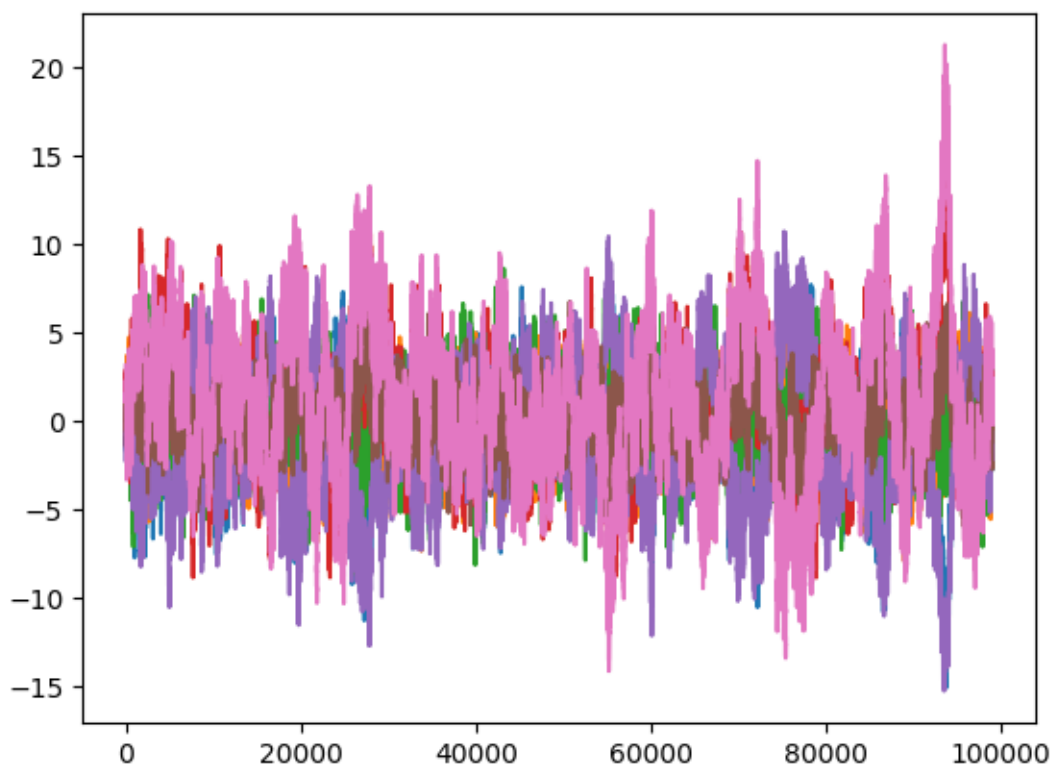
```

```

PyObject <matplotlib.lines.Line2D object at 0x00000000548F8AC8>
PyObject <matplotlib.lines.Line2D object at 0x00000000548F8C88>
PyObject <matplotlib.lines.Line2D object at 0x00000000548F8E48>
PyObject <matplotlib.lines.Line2D object at 0x00000000548FF148>
PyObject <matplotlib.lines.Line2D object at 0x00000000548FF408>

```

```
[36]: plot(Y[:,1:10:end'],'')
```



```

[36]: 7-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x000000005E965E08>
PyObject <matplotlib.lines.Line2D object at 0x000000005E9709C8>
PyObject <matplotlib.lines.Line2D object at 0x000000005E970B88>
PyObject <matplotlib.lines.Line2D object at 0x000000005E970D48>
PyObject <matplotlib.lines.Line2D object at 0x000000005E970F48>
PyObject <matplotlib.lines.Line2D object at 0x000000005E9771C8>
PyObject <matplotlib.lines.Line2D object at 0x000000005E9773C8>

```

```

[38]: noise_dist = MvNormal(h*I + zeros(size(Y,1),size(Y,1)))

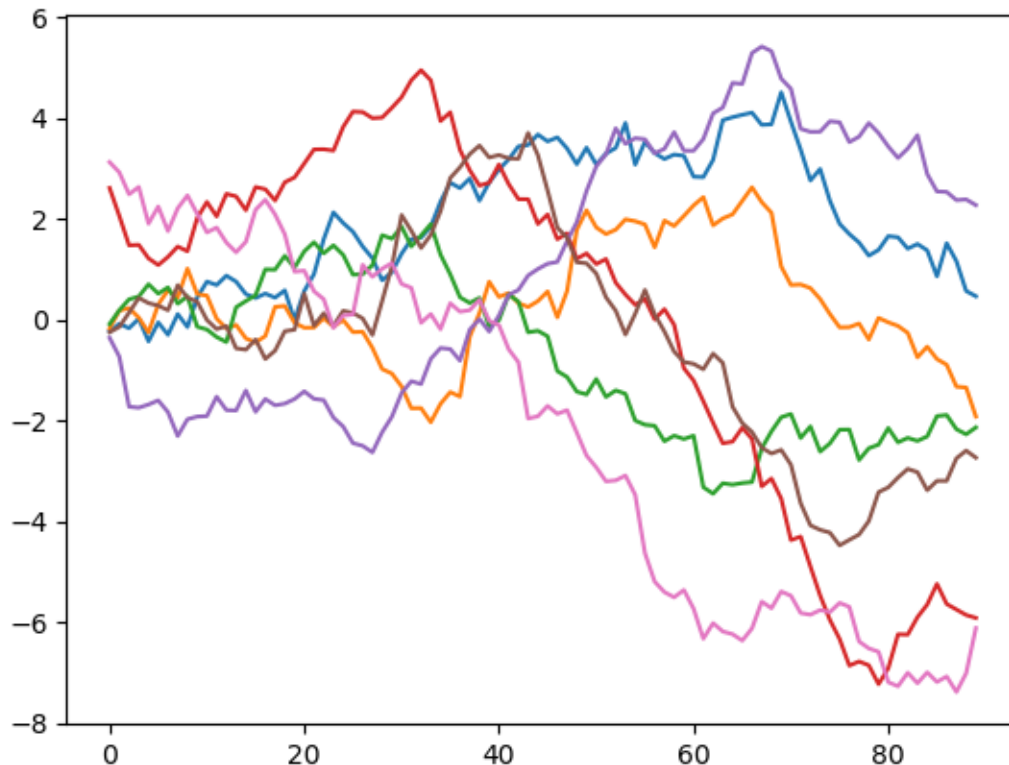
Y_rm = redmodrun(Y, h_wf_all, Psi;noise_dist)

```

reduced model blewup at step 981024

```
[38]: 7×990001 Array{Complex{Float64},2}:
      -0.22646+0.0im -0.182972+0.0im ... 0.0+0.0im 0.0+0.0im 0.0+0.0im
      -0.162249+0.0im -0.201968+0.0im 0.0+0.0im 0.0+0.0im 0.0+0.0im
      -0.0760985+0.0im -0.107251+0.0im 0.0+0.0im 0.0+0.0im 0.0+0.0im
      2.62242+0.0im 2.58614+0.0im 0.0+0.0im 0.0+0.0im 0.0+0.0im
      -0.349861+0.0im -0.488975+0.0im 0.0+0.0im 0.0+0.0im 0.0+0.0im
      -0.239206+0.0im -0.373286+0.0im ... 0.0+0.0im 0.0+0.0im 0.0+0.0im
      3.13468+0.0im 3.10144+0.0im 0.0+0.0im 0.0+0.0im 0.0+0.0im
```

```
[43]: plot(Y_rm[:,1:10:900]')
```

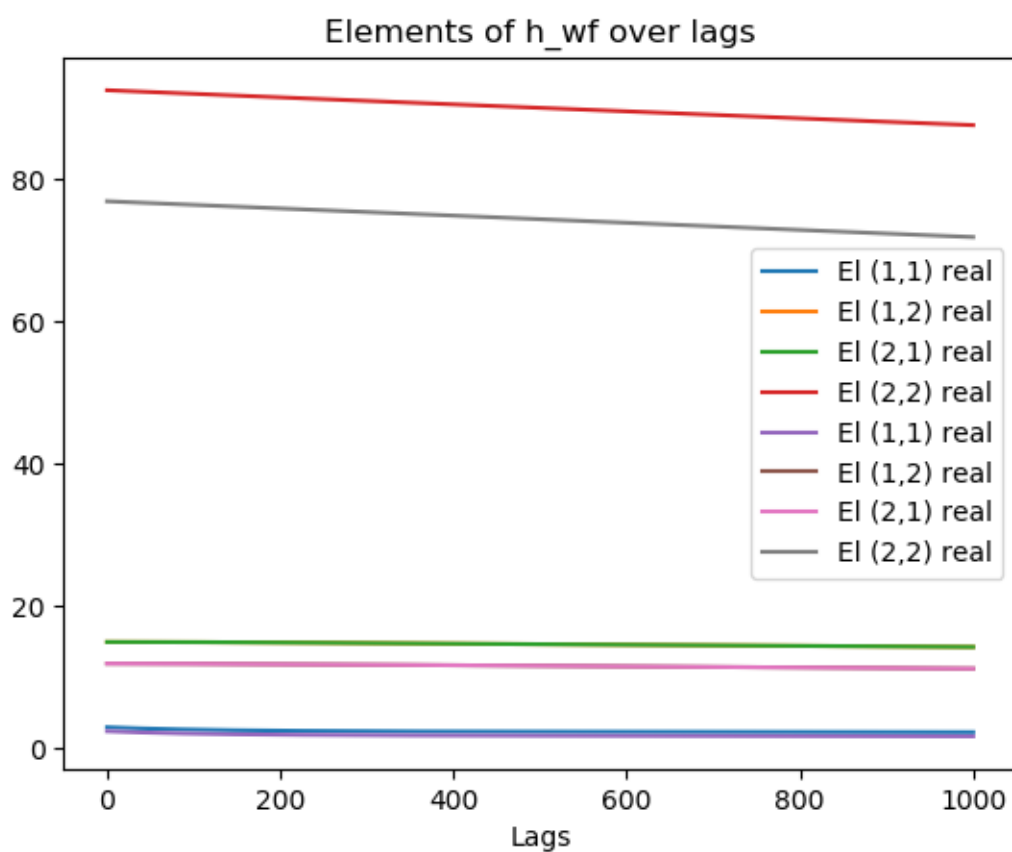


```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
  return array(a, dtype, copy=False, order=order)
```

```
[43]: 7-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x000000005EA6FC88>
PyObject <matplotlib.lines.Line2D object at 0x000000005EA7A848>
PyObject <matplotlib.lines.Line2D object at 0x000000005EA7AA08>
PyObject <matplotlib.lines.Line2D object at 0x000000005EA7ABC8>
PyObject <matplotlib.lines.Line2D object at 0x000000005EA7ADC8>
PyObject <matplotlib.lines.Line2D object at 0x000000005EA80048>
```

PyObject <matplotlib.lines.Line2D object at 0x000000005EA80248>

```
[171]: C = matrix_autocov_seq(Y;  
    L = 1000, win = "par"  
    )  
  
C_rm = matrix_autocov_seq(Y_rm;  
    L = 1000, win = "par"  
    )  
  
plot_WF(C)  
plot_WF(C_rm)
```



```
[171]: PyObject Text(0.5, 1, 'Elements of h_wf over lags')
```

1.8 Model reduction

```
[44]: # Put in Psi functions
Psi(x) = x

# Model reduction Parameters
M_out = 500
n = 2
p = 500
par = 55
ty = "bin"
rl = true
Preds = true
PI = false
rtol = 1e-6

nfft = 2^10

@time h_wf_1, pred = get_wf(Y[1:1,:], Psi;
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);

Y_sig_1 = Y[1:1,2:end]
```

Number of CKMS iterations: 1292

errK errR : 8.314911948558373e-11 1.056593669053811e-17

1.354334 seconds (3.83 M allocations: 847.112 MiB, 5.99% gc time)

```
[44]: 1×990000 Array{Float64,2}:
 -0.182972 -0.218403 -0.203021 ... -0.920501 -0.987486 -0.953155
```

```
[45]: h_wf_1
```

```
[45]: 1×1×500 Array{Float64,3}:
```

```
[:, :, 1] =
```

```
0.9979884299727003
```

```
[:, :, 2] =
```

```
0.001055247416671034
```

```
[:, :, 3] =
```

```
-0.0007628300176600088
```

```
...
```

```
[:, :, 498] =
```

```
0.0001347367145882749
```

```
[:, :, 499] =
```



```
-0.00023991659756247046
```

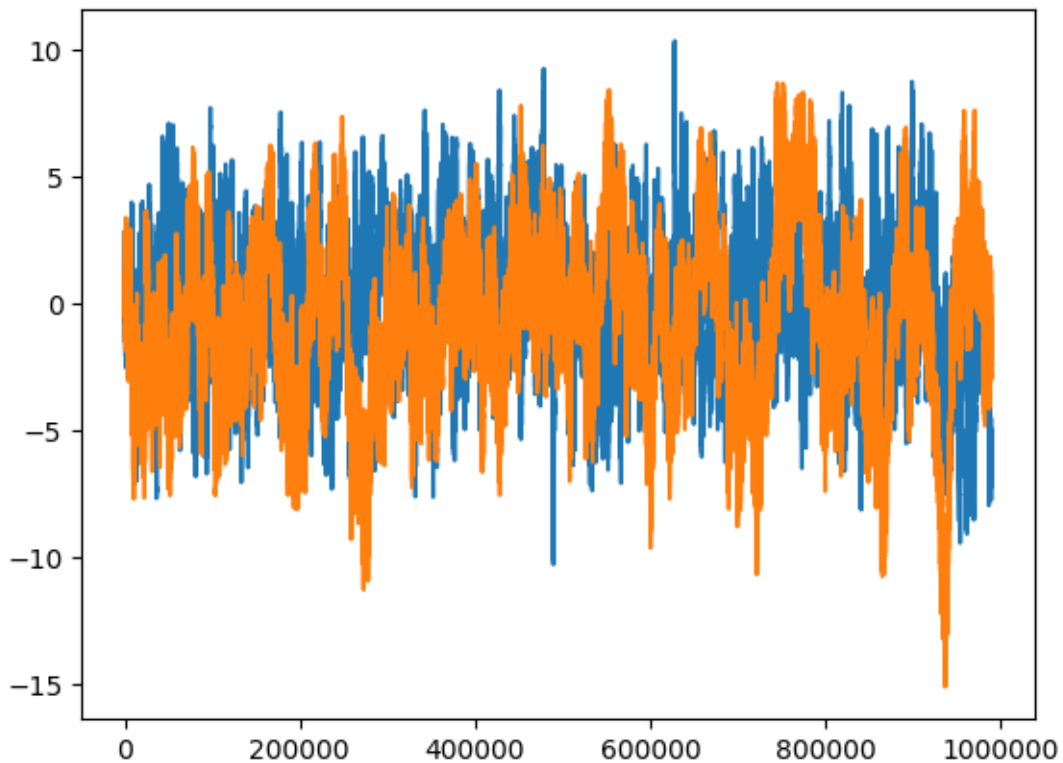
```
[:, :, 500] =  
-9.848363775375039e-5
```

```
[46]: noise_dist = MvNormal(h*I + zeros(1,1))  
  
Y_rm_1 = redmodrun(real(Y[1:1,:]), h_wf_1, Psi;noise_dist)
```

reduced model did not blowup

```
[46]: 1×990001 Array{Complex{Float64},2}:  
-0.22646+0.0im -0.182972+0.0im ... -5.97856+0.0im -6.11601+0.0im
```

```
[47]: plot([Y_rm_1[:, :] Y[1,:]])
```

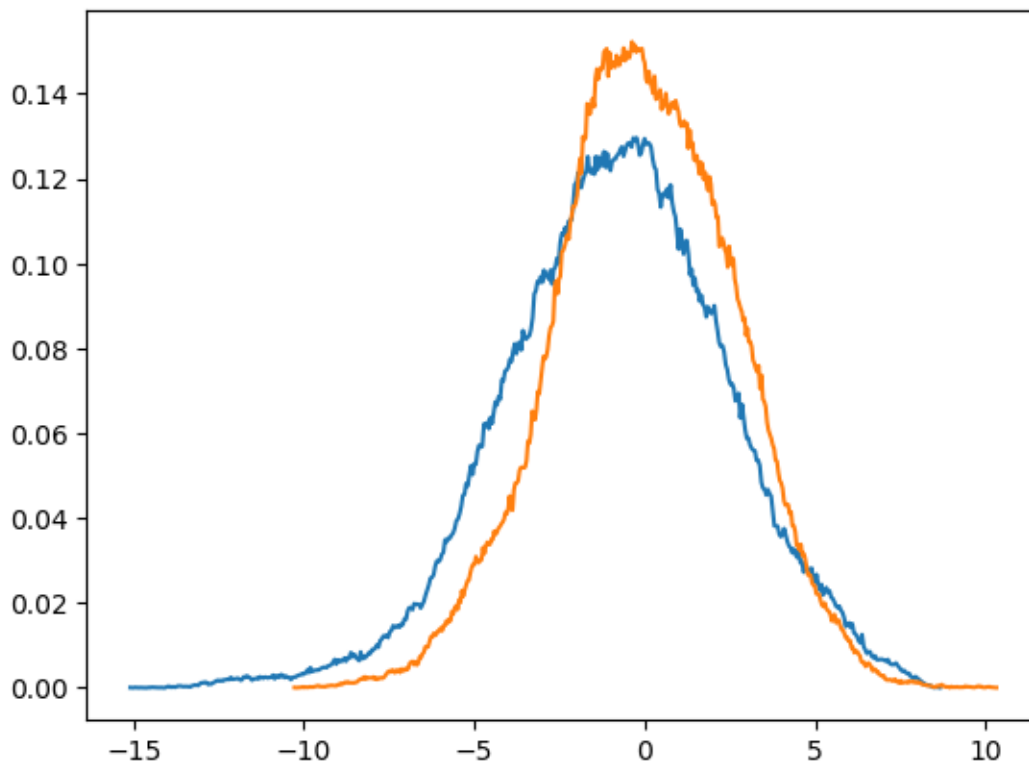


```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:  
ComplexWarning: Casting complex values to real discards the imaginary part  
return array(a, dtype, copy=False, order=order)
```

```
[47]: 2-element Array{PyCall.PyObject,1}:  
PyObject <matplotlib.lines.Line2D object at 0x00000000627A6C08>
```

PyObject <matplotlib.lines.Line2D object at 0x00000000627B07C8>

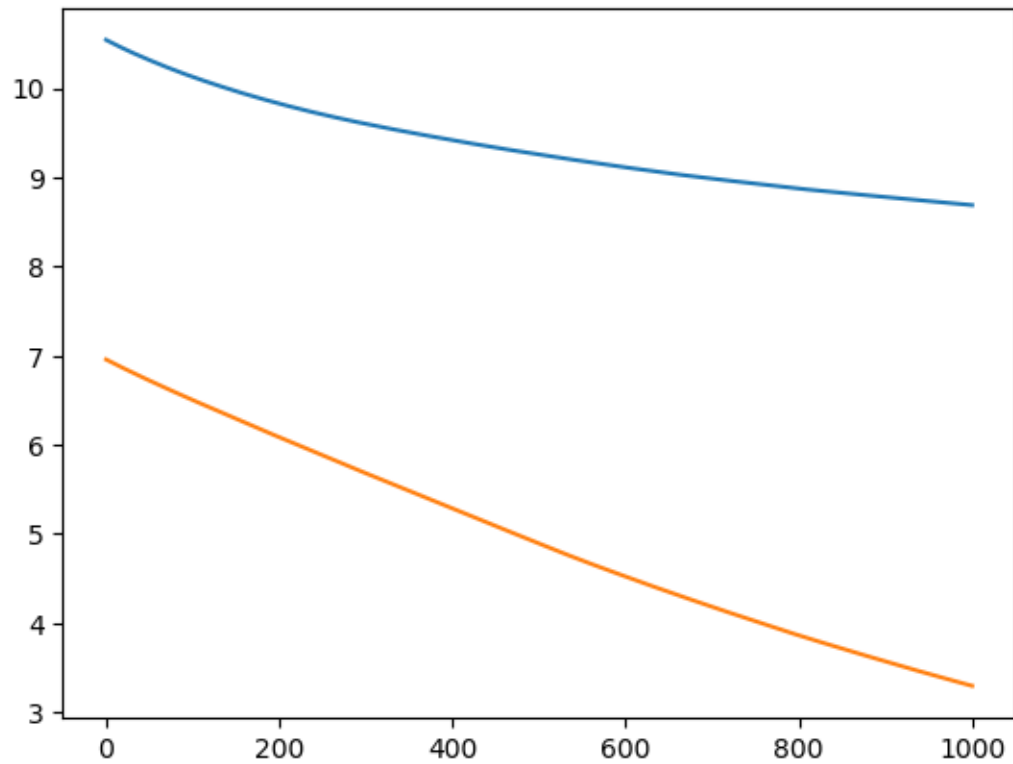
```
[48]: emp_pdf(real(Y[1,:]),bn = 500)
      emp_pdf(real(Y_rm_1[:]),bn = 500)
```



```
[48]: 3-element Array{Array{T,1} where T,1}:
 [7.343833403637794e-5, 2.447944467879264e-5, 4.89588893575853e-5,
 0.00012239722339396324, 7.343833403637791e-5, 0.0001223972233939632,
 0.0002447944467879265, 9.791777871517052e-5, 0.00017135611275154858,
 0.0001223972233939632 ... 0.00012239722339500798, 7.343833403646667e-5,
 0.00019583555742878406, 0.0002203150021094, 0.00026927389146794134,
 0.0001223972233923174, 0.00014687666807293334, 7.343833403646667e-5,
 4.895888935854131e-5, 4.895888935854131e-5]
 [-10.280445834712623, -10.23918264418473, -10.197919453656837,
 -10.156656263128946, -10.115393072601055, -10.074129882073162,
 -10.03286669154527, -9.991603501017378, -9.950340310489487, -9.909077119961594
 ... 9.938517523954427, 9.979780714482319, 10.021043905010211, 10.062307095538104,
 10.103570286065995, 10.144833476593886, 10.18609666712178, 10.227359857649672,
 10.268623048177563, 10.309886238705454]
 PyCall.PyObject[PyObject <matplotlib.lines.Line2D object at
 0x00000000631C6A48>]
```

```
[50]: lags = 0:1000
A_1 = my_autocov(real(Y[1,:]),lags)
A_rm_1 = my_autocov(real(Y_rm_1[:]),lags)

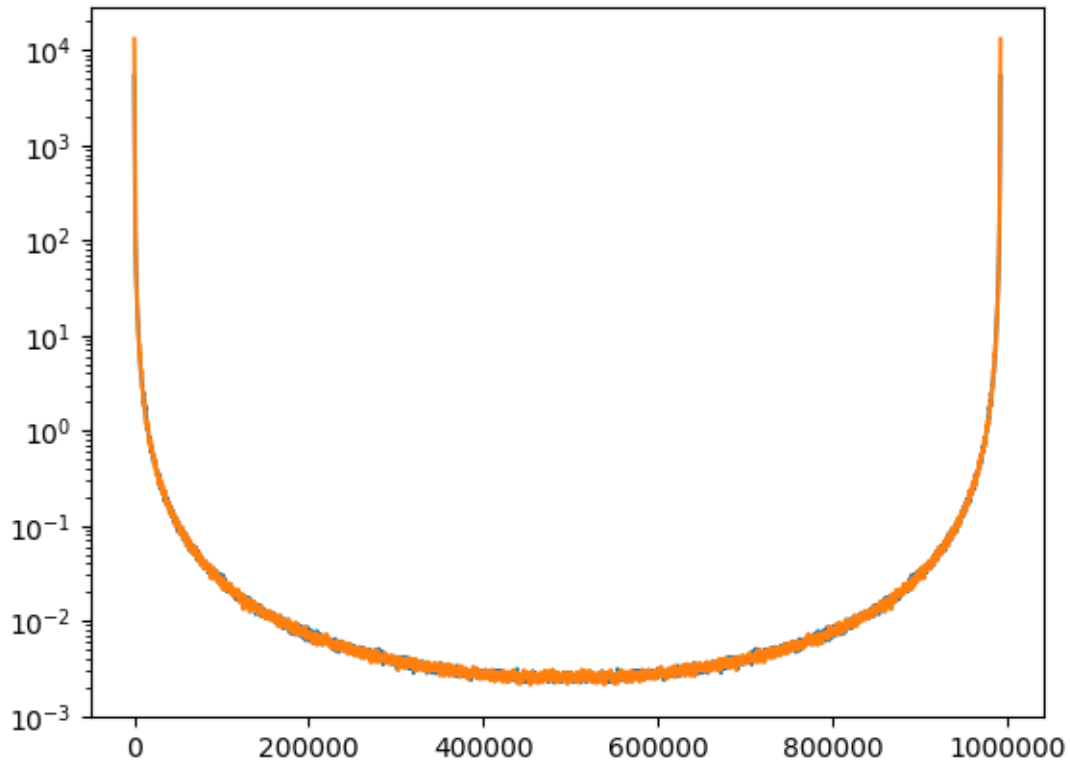
plot([A_1 A_rm_1])
```



```
[50]: 2-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x00000000636787C8>
PyObject <matplotlib.lines.Line2D object at 0x0000000063683388>
```

```
[52]: S_1 = z_crossspect_scalar(real(Y[2,:]),real(Y[2,:]))
S_rm_1 = z_crossspect_scalar(real(Y_rm_1[1,:]),real(Y_rm_1[1,:]))

semilogy([S_1 S_rm_1])
```



C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)

[52]: 2-element Array{PyCall.PyObject,1}:
PyObject <matplotlib.lines.Line2D object at 0x000000006683E408>
PyObject <matplotlib.lines.Line2D object at 0x000000006B261D88>

```
[53]: # Put in Psi functions
Psi(x) = x

# Model reduction Parameters
M_out = 100
n = 2
p = 500
par = 55
ty = "bin"
rl = true
Preds = true
PI = false
rtol = 1e-6
```

```

nfft = 1024

@time h_wf_2, pred = get_wf(Y[2:2,:], Psi;
    M_out, n, p, par, ty, nfft, rl, Preds, PI, rtol);

Y_sig_2 = Y[:,2:end]

```

Number of CKMS iterations: 187
 errK errR : 6.193752338281715e-11 1.0970249493569364e-14
 0.779706 seconds (1.10 M allocations: 506.558 MiB, 2.86% gc time)

```

[53]: 7×990000 Array{Float64,2}:
 -0.182972 -0.218403 -0.203021 ... -0.920501 -0.987486 -0.953155
 -0.201968 -0.0401256 0.0412264 -0.634519 -0.670849 -0.655005
 -0.107251 -0.190555 -0.115105 1.17574 1.01653 1.0076
 2.58614 2.35267 2.41436 3.1203 3.24898 3.20971
 -0.488975 -0.513428 -0.455456 -0.679365 -0.64448 -0.669547
 -0.373286 -0.434784 -0.417762 ... -2.51162 -2.39705 -2.40973
 3.10144 3.03561 2.95586 -0.167722 -0.270254 -0.190301

```

```
[54]: h_wf_2
```

```

[54]: 1×1×100 Array{Float64,3}:
[:, :, 1] =
 0.9989819922461443

[:, :, 2] =
 -0.0015097310402257826

[:, :, 3] =
 1.3274735809500212e-5

...

[:, :, 98] =
 1.9353317376827833e-6

[:, :, 99] =
 1.0836565379960829e-6

[:, :, 100] =
 -7.037668765354446e-8

```

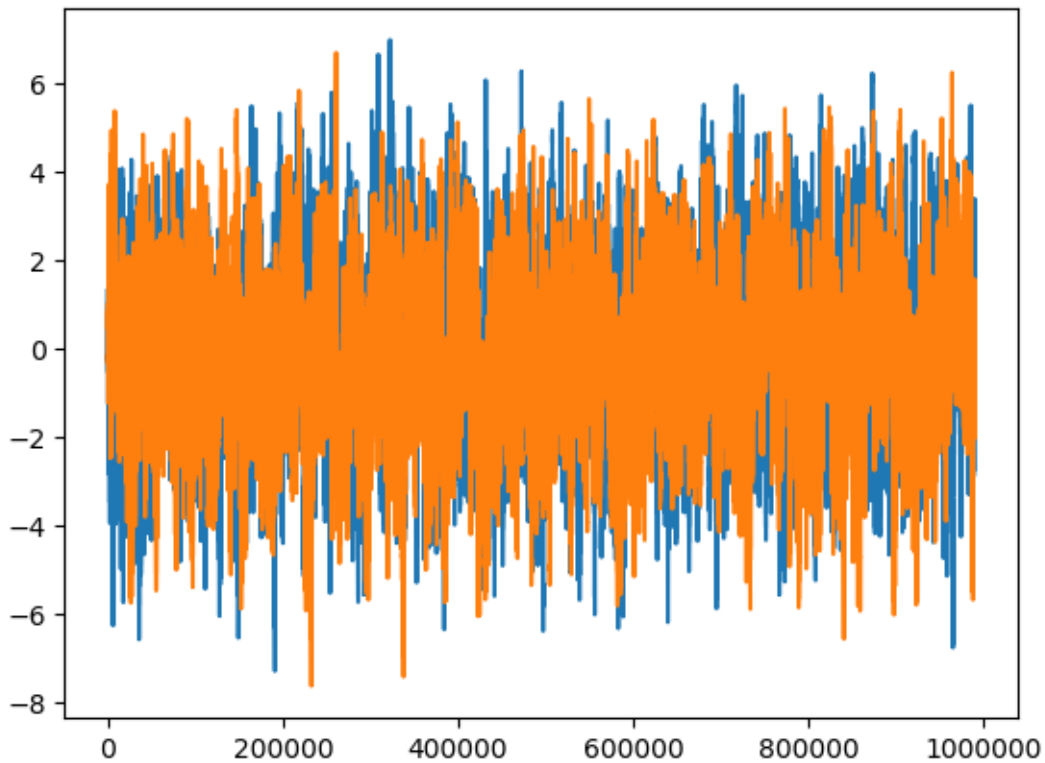
```

[55]: noise_dist = MvNormal(h*I + zeros(1,1))

Y_rm_2 = redmodrun(real(Y[2:2,:]), h_wf_2, Psi;noise_dist)

```

```
plot([Y_rm_2[:] Y[2,:]])
```

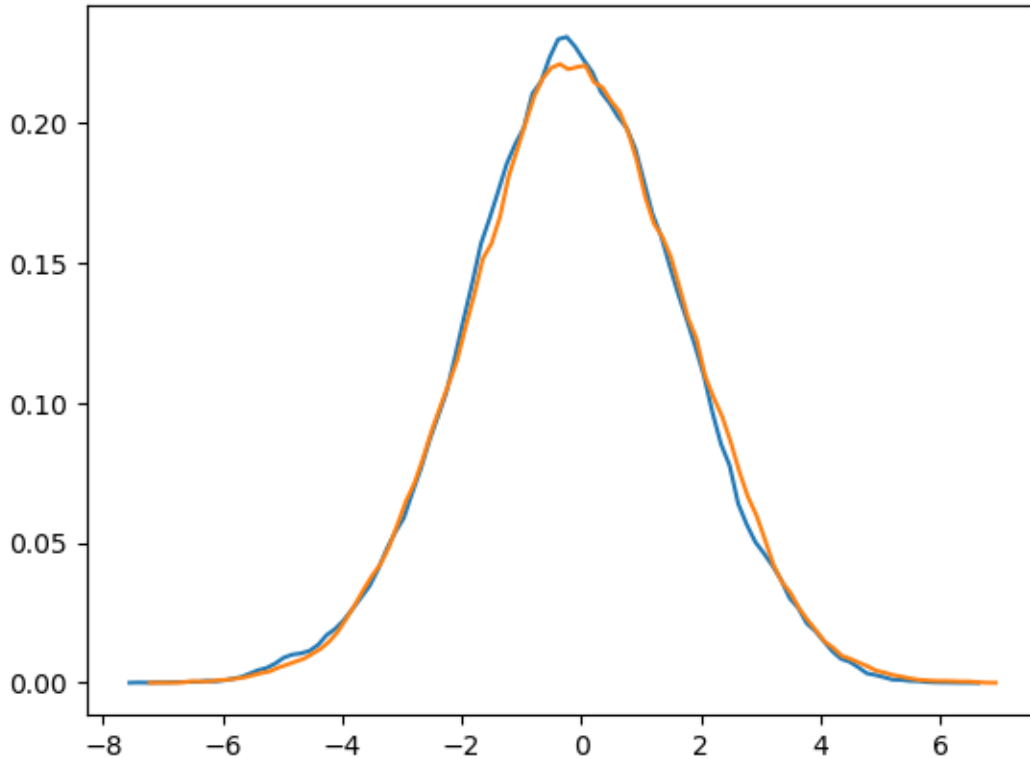


reduced model did not blowup

```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:  
ComplexWarning: Casting complex values to real discards the imaginary part  
return array(a, dtype, copy=False, order=order)
```

```
[55]: 2-element Array{PyObject,1}:  
      PyObject <matplotlib.lines.Line2D object at 0x0000000065FD9D08>  
      PyObject <matplotlib.lines.Line2D object at 0x0000000065FD1488>
```

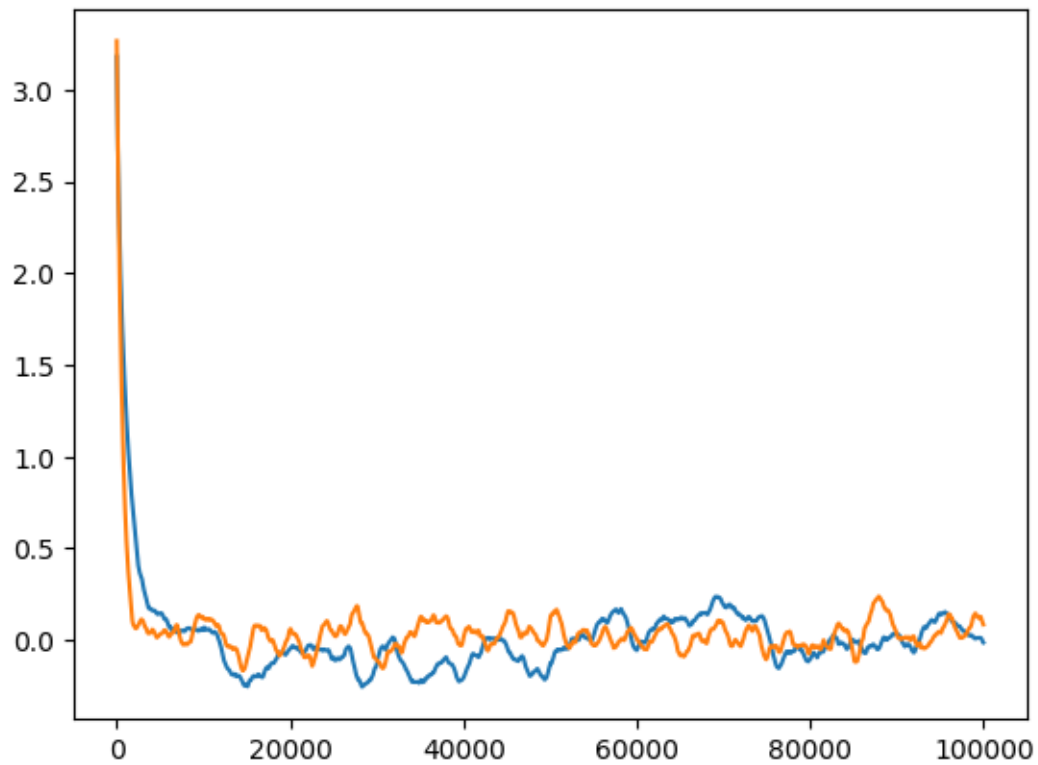
```
[56]: emp_pdf(real(Y[2,:]),bn = 100)  
      emp_pdf(real(Y_rm_2[:]),bn = 100)
```



```
[56]: 3-element Array{Array{T,1} where T,1}:
 [3.534553428602587e-5, 7.069106857205173e-5, 5.655285485764139e-5,
 5.6552854857641404e-5, 0.0002191423125733604, 0.0006079431897196449,
 0.0005160448005759777, 0.0005725976554336192, 0.0009331221051510829,
 0.0011098497765812123 ... 0.0010886424560098763, 0.0009048456777220606,
 0.0007139797925780673, 0.0007069106857203356, 0.000622081403433771,
 0.0005089756937193875, 0.0004877683731469694, 0.00021207320571641148,
 7.069106857187816e-5, 4.2414641143282295e-5]
 [-7.218055141920095, -7.075165805232752, -6.932276468545409,
 -6.789387131858066, -6.646497795170724, -6.503608458483381, -6.360719121796038,
 -6.217829785108696, -6.074940448421353, -5.93205111173401 ... 5.641985159940745,
 5.784874496628087, 5.9277638333154306, 6.070653170002774, 6.213542506690116,
 6.356431843377458, 6.499321180064801, 6.642210516752144, 6.785099853439486,
 6.927989190126829]
 PyCall.PyObject[PyObject <matplotlib.lines.Line2D object at
 0x000000006CB7BFC8>]
```

```
[57]: lags = 0:100000
A_2 = my_autocov(real(Y[2,:]),lags)
A_rm_2 = my_autocov(real(Y_rm_2[:]),lags)

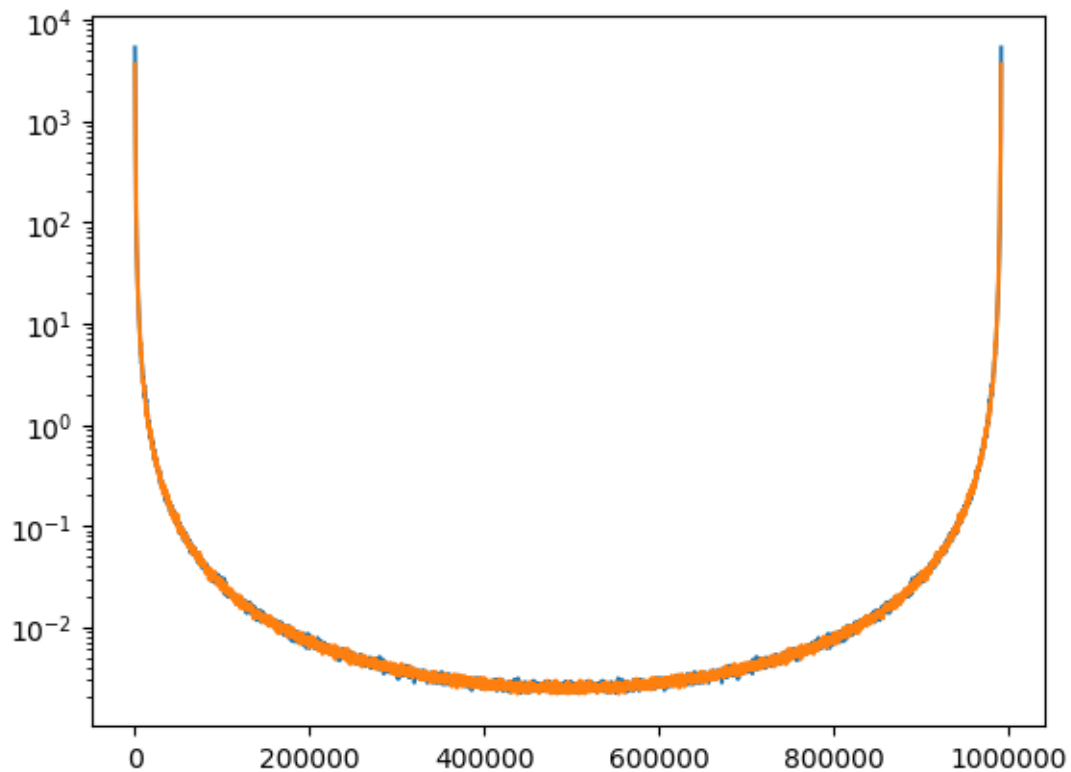
plot([A_2 A_rm_2])
```



```
[57]: 2-element Array{PyCall.PyObject,1}:
      PyObject <matplotlib.lines.Line2D object at 0x000000006B274048>
      PyObject <matplotlib.lines.Line2D object at 0x000000006CBB8448>
```

```
[59]: S_2 = z_crossspect_scalar(real(Y[2,:]),real(Y[2,:]))
      S_rm_2 = z_crossspect_scalar(real(Y_rm_2[1,:]),real(Y_rm_2[1,:]))

      semilogy([S_2 S_rm_2])
```

```
C:\Users\jared\.julia\conda\3\lib\site-packages\numpy\core\_asarray.py:85:
ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)
```

```
[59]: 2-element Array{PyCall.PyObject,1}:
      PyObject <matplotlib.lines.Line2D object at 0x000000006E8B0388>
      PyObject <matplotlib.lines.Line2D object at 0x000000006E8BFC08>
```

```
[ ]:
```