Tutorial 6: DMFT calculations for the Hubbard model, metal-insulator transition, KLM, PAM

Rok Žitko
Institute Jožef Stefan
Ljubljana, Slovenia

Updated lecture & tutorial slides, and update materials for tutorials:

http://nrgljubljana.ijs.si/

Documentation and examples

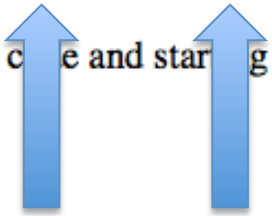
The reference manual is now being prepared. Here's the current version: (version from May 30, 2013). An old version (2007) of the manual is also availble. It is quite outdated, but provides some potentially useful background information, which has not been integrated into the new reference manual yet.

A Mathematica implementation of the NRG method is also available. It illustrates the main ideas of the algorithm on the single-impurity Anderson model using simple Mathematica notebook interface. It calculates the thermodynamic quantities and the expectation values of arbitrary local operators.

Also, check out the examples and tutorials.

NEW: NRG tutorials at SISSA, June 2013: slides (ppt, pdf) and tutorials.

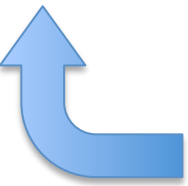
(Note: 45MB compressed.)



I'm happy to assist prospective users in setting up the c e and star g to do first calculations, so don't hesitate to contact me via e-mail.

Implementation using shell and perl scripts

- **dmft_mon**:
  - initializes the calculation (**`dmft_init`**, builds an approximation for Δ)
  - runs `nrginit` and submits the calculations (`nrgrun`) to the cluser
  - waits for the calculations to finish, processes the results (**`dmft_done`**)
  - checks for convergence (**`checkconv`**): if not, starts a new loop



- **`dmft_done`**:
  - **`average`**: do the z-averaging of spectral functions
  - **`realparts`**: Kramers-Kronig to obtain real parts of G and F
  - **`sigmatrick`**: compute Σ
  - **`bandDOS`**: compute lattice G
  - **`copyresults`** (snapshot of current DMFT step), occupancies (<n>, etc.)
  - **`dmftDOS`**: compute Δ for new DMFT step
  - **`broyden`**: do the Broyden mixing in order to obtain improved Δ

param.loop

```
#!/usr/bin/env looper
#
#PRELUDE: $Nz=4; $DELTA=`cat param.delta`;
#AUTOLOOP: ../odesolv1 ; nrginit ; nrgrun

[dmft]
Nz=4
# D=1/scale=0.1, thus W=2D=0.2
scale=10
clip=1e-4

# Target occupancy
goal=0.8

# Broyden
offset=1
M=1000
alpha=0.5

[param]
xmax=15
adapt=false

tri=cpp
preccpp=2000
prec=100

symtype=QS
model=../model.m

fixeps=1e-10
```

`param.delta` contains δ, the shift of the center of the conduction band (note than μ=0)

discretization solver

we need to rescale all the energies to make them fit in the NRG energy window of [-1:1]

fixed occupancy calculation with <n>=0.8

parameters for odesolv

parameters for tridiagonalization

```
U=0.22

delta=$DELTA

Lambda=4.0
Tmin=1e-10
keepmin=500
keepenergy=10.0
keep=8000

discretization=Z
band=asymode
dos=../Delta.dat

@$z = 1.0/$Nz; $z <= 1.00001; $z += 1/$Nz
z=$z

strategy=kept
ops=A_d self_d n_d n_d_ud Himp Hhyb Hpot
specd=A_d-A_d self_d-A_d
```

0.22 x scale -> U=2.2D

arbitrary DOS

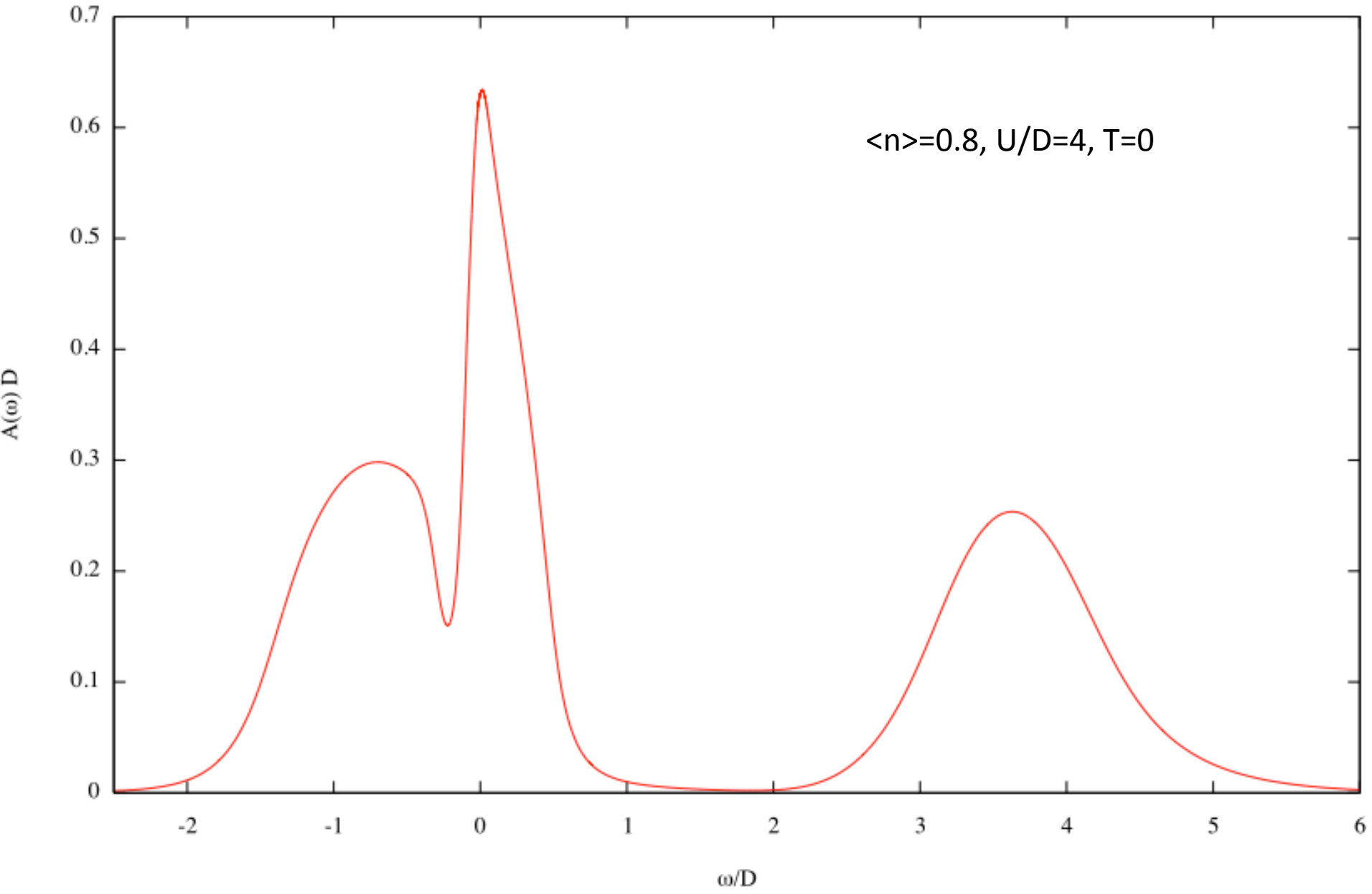
```
fdm=true
fdmexpv=true

broaden=false
savebins=true
bins=1000

T=1e-09

broaden_max=4
broaden_ratio=1.01
broaden_min=1e-8

width_custom=20
prec_custom=14

done=true
```

model.m

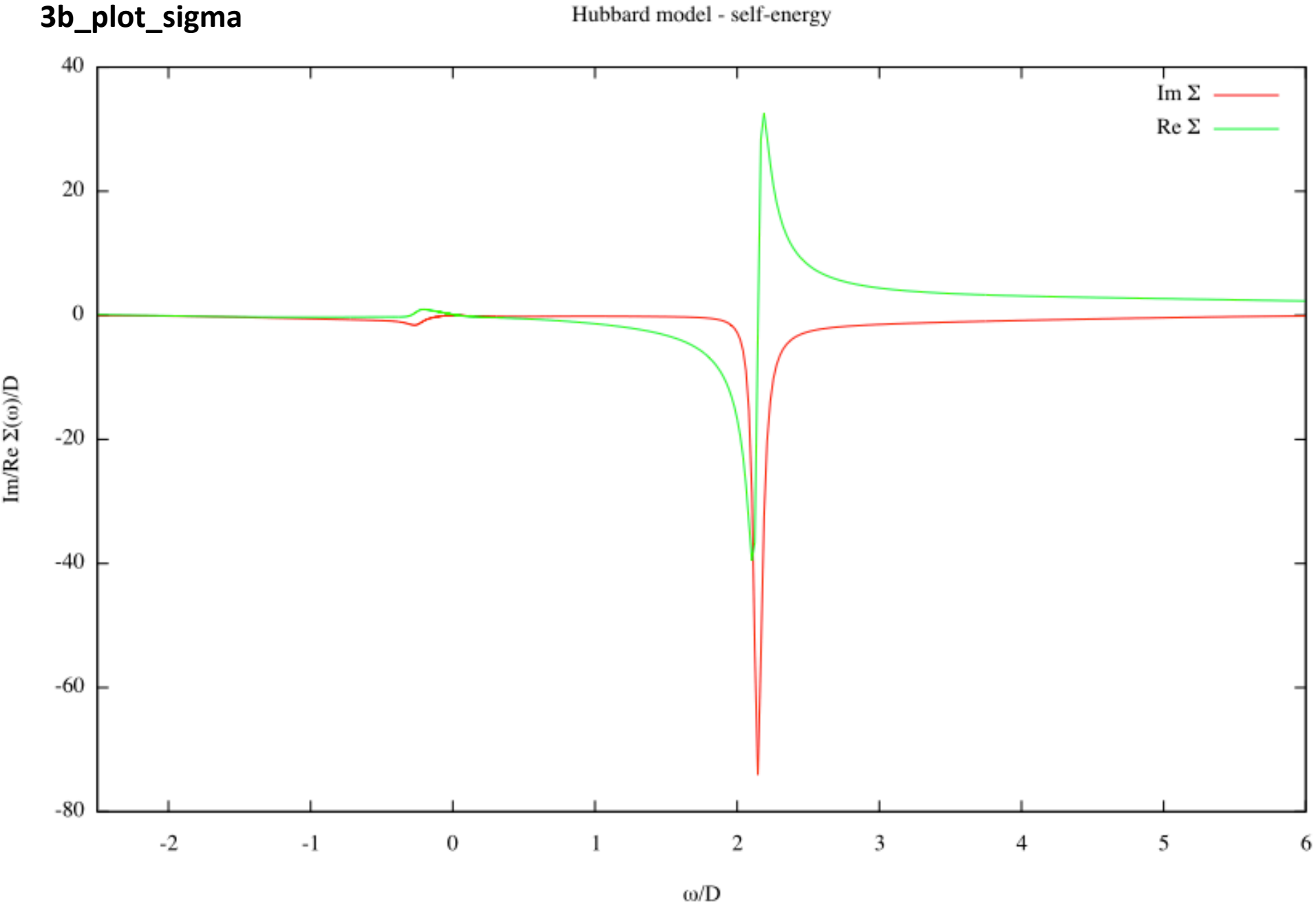
```
def1ch[1];

H1 = delta number[d[]] + U/2 pow[number[d[]]-1, 2];
Hc = Sum[gammaPolCh[ch] hop[f[ch-1], d[]], {ch, CHANNELS}];

Himp = H1;
Hhyb = Hc;
Hpot = U hubbard[d[]];

H = Himp + Hhyb + H0;

(* All operators which contain d[], except hybridization (Hc). *)
Hselfd = Himp;

selfopd = ( Chop @ Expand @ komutator[Hselfd /. params, d[#1, #2]] )&;
```

3a_plot

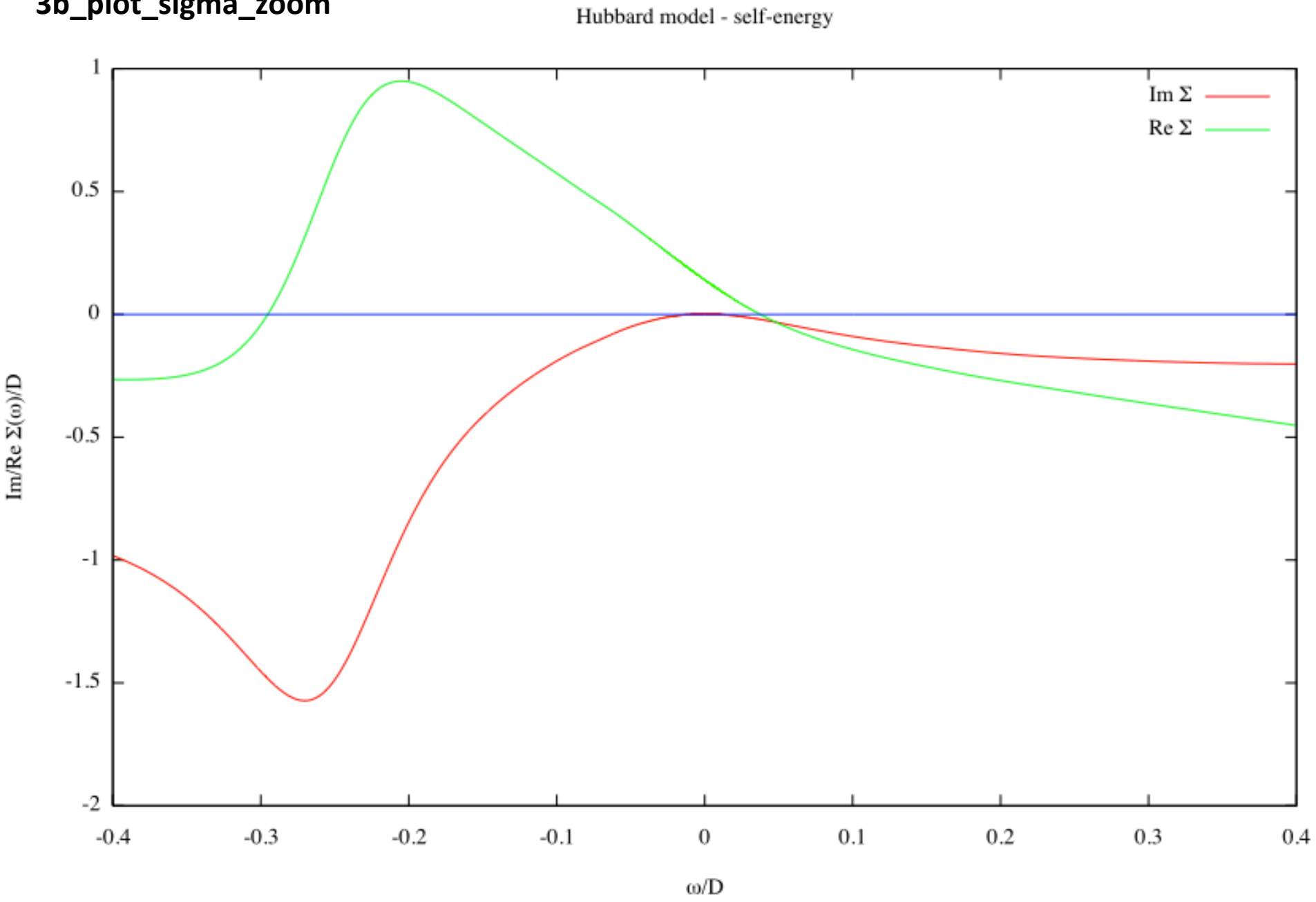
45_Hubbard



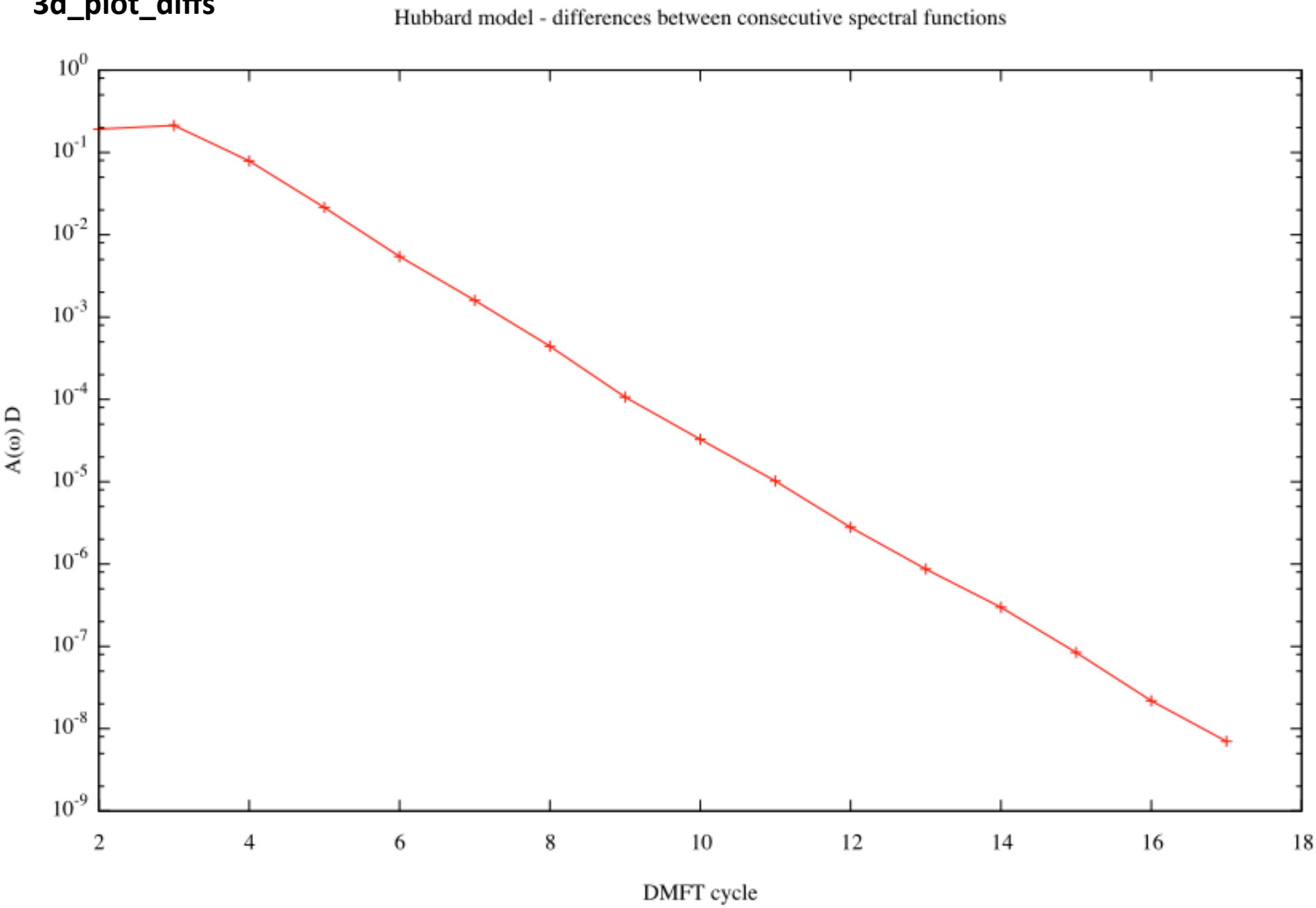
3b_plot_sigma



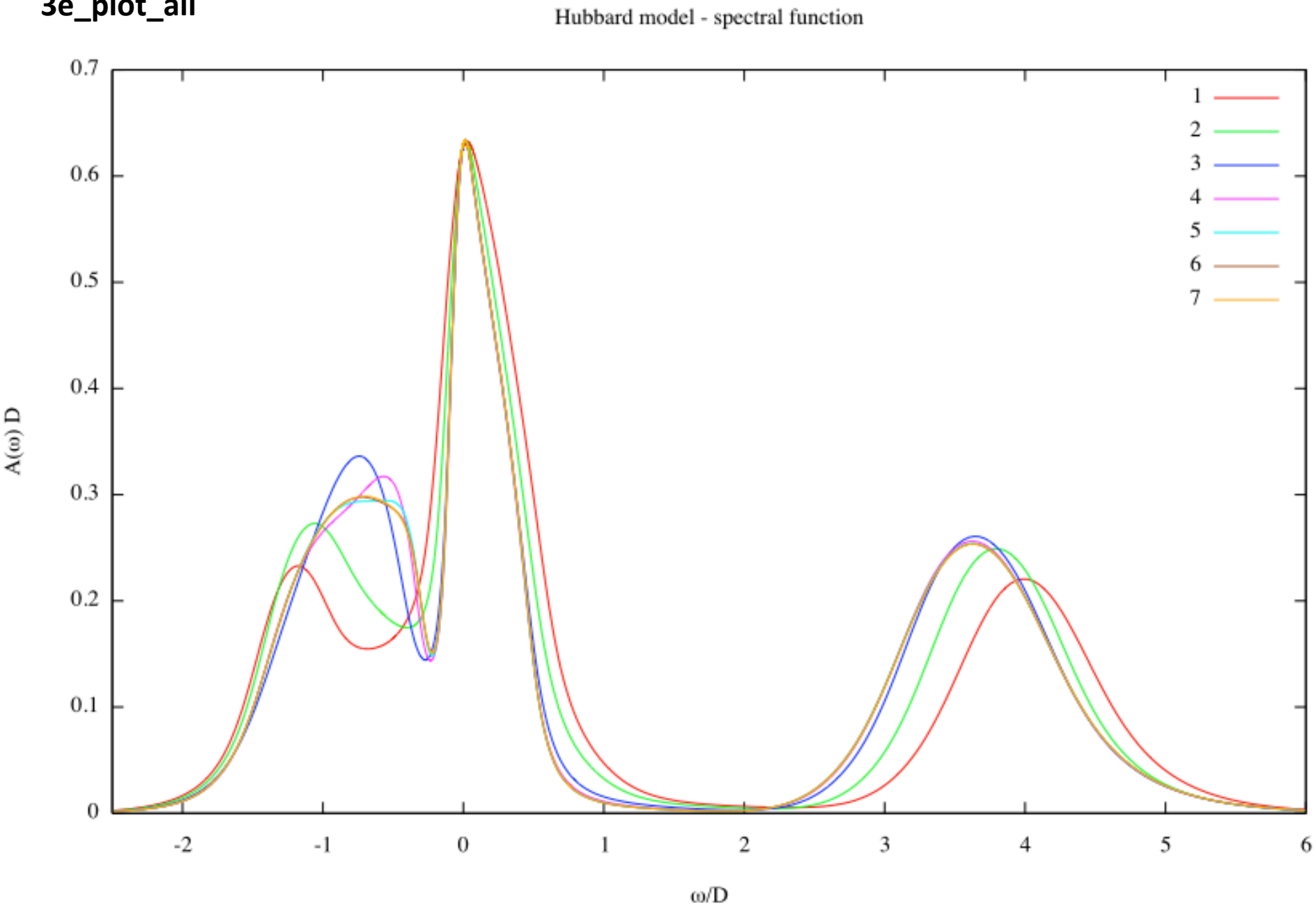
3b_plot_sigma_zoom

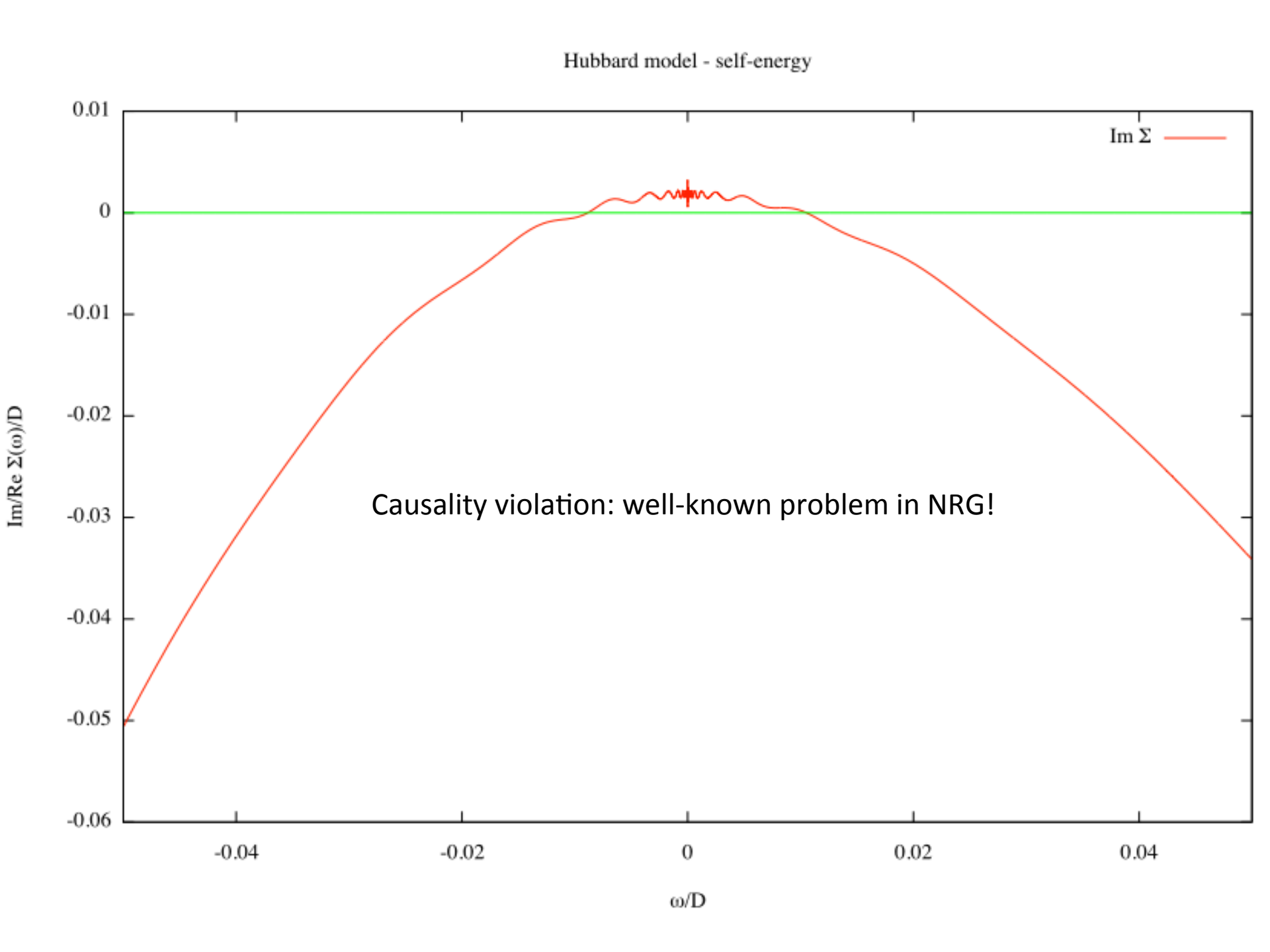


3d_plot_diffs



3e_plot_all


Hubbard model - spectral function
A(ω) D
ω/D
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
-2
-1
0
1
2
3
4
5
6
1
2
3
4
5
6
7


Hubbard model - self-energy
Im Σ
0.01
0
-0.01
-0.02
-0.03
-0.04
-0.05
-0.06
Im/Re Σ(ω)/D
-0.04
-0.02
0
0.02
0.04
ω/D
Causality violation: well-known problem in NRG!

NOTE: when restarting a DMFT calculation, erase the file `ITER`. This tells the code to restart (and prevents Broyden mixer from complaining that the previous results cannot be loaded). This will still reuse the result from the previous calculation as an initial approximation.

If you want to start from scratch, erase the file `ITER` and all files named `Delta*`.

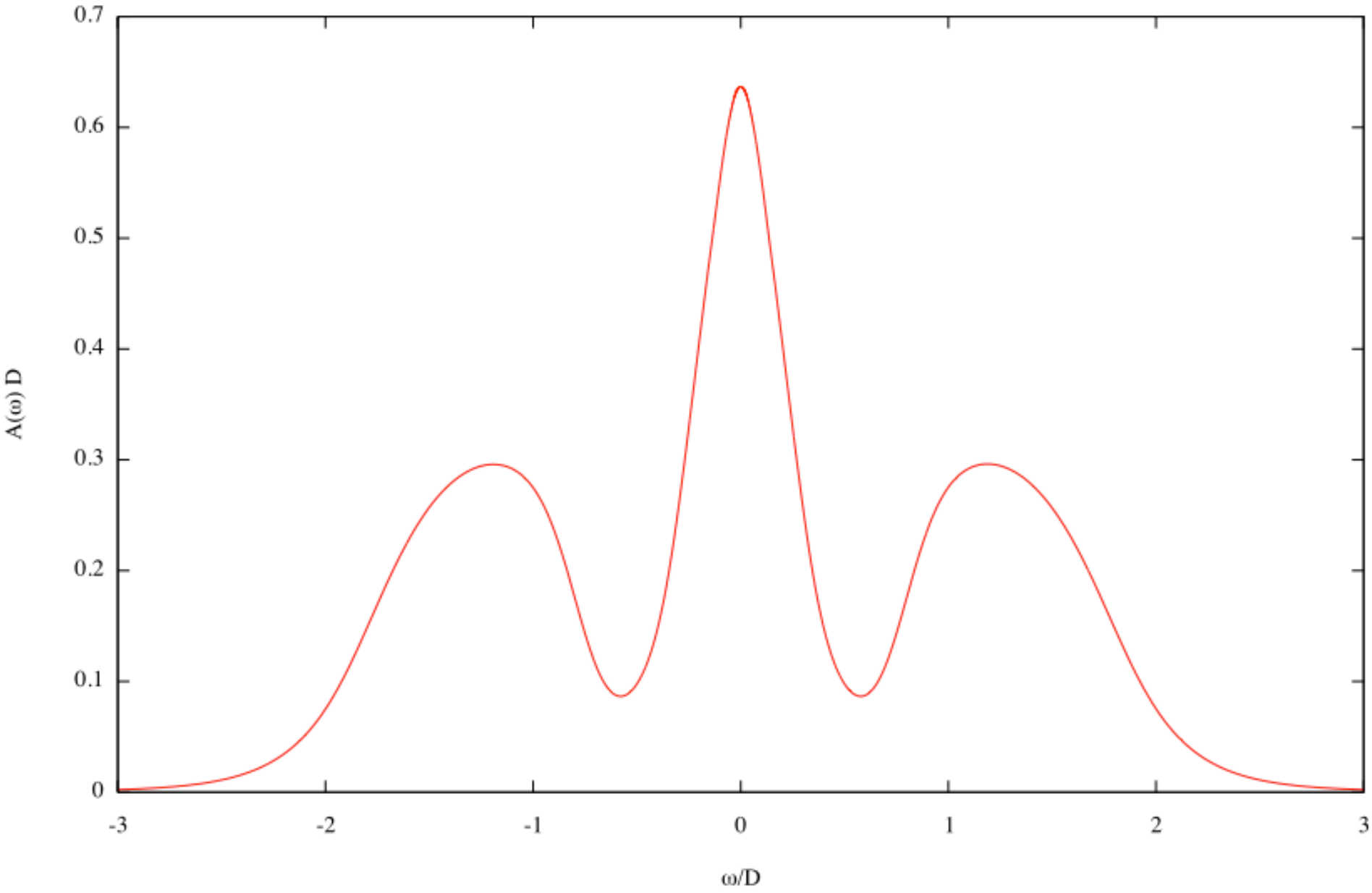
Exercises

- Reduce the doping (i.e., goal $\rightarrow$ 1). How does the quasiparticle peak evolve?
- Increase the temperature. Follow the changes in the spectral function and in Σ.
- Try different initial hybridization functions (i.e., create a file `DeltaFirst.dat`). Is the converged solution always the same?

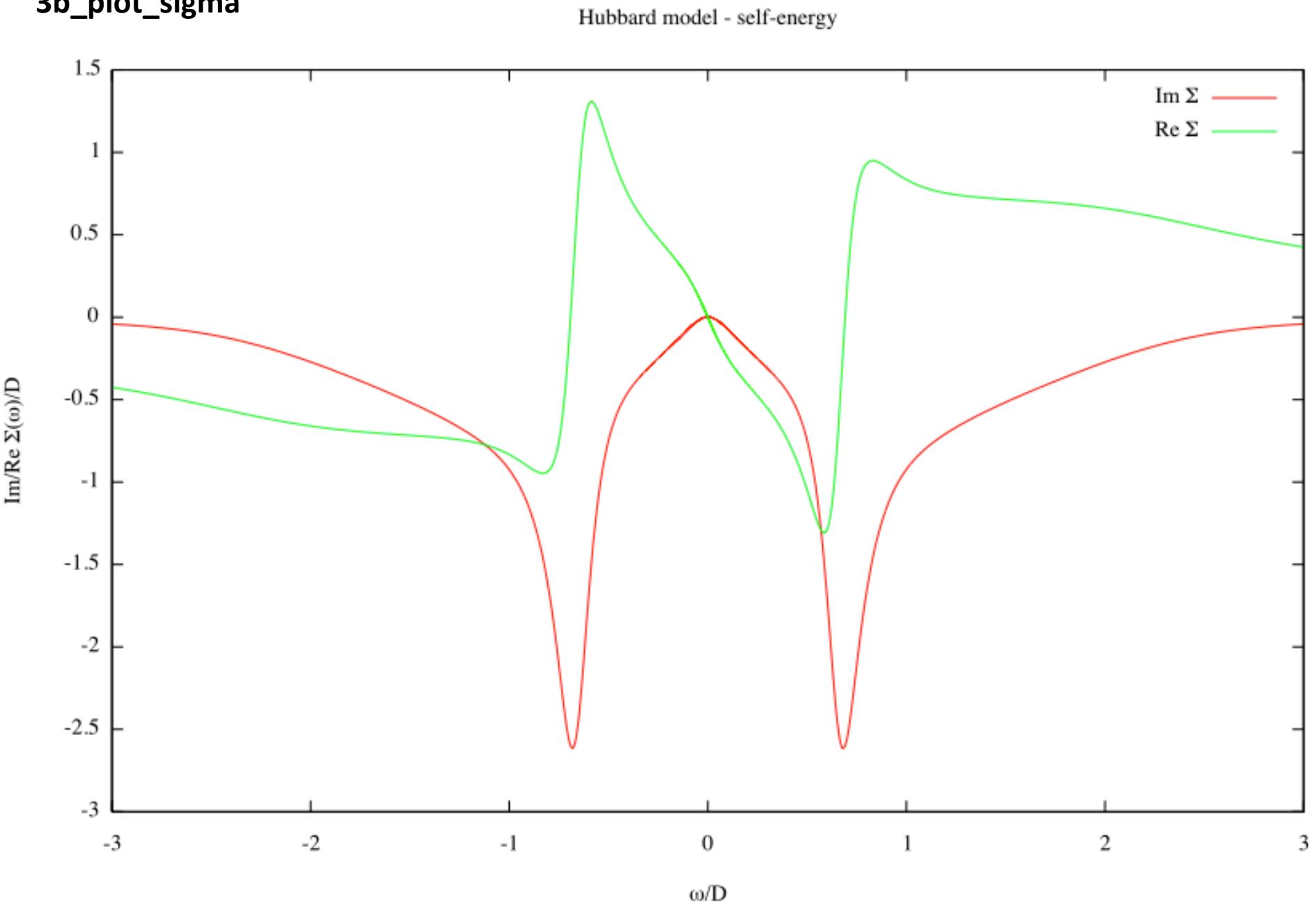
Note: see also **47_Hubbard_fast**, larger Λ=4, much faster calculations!

3a_plot

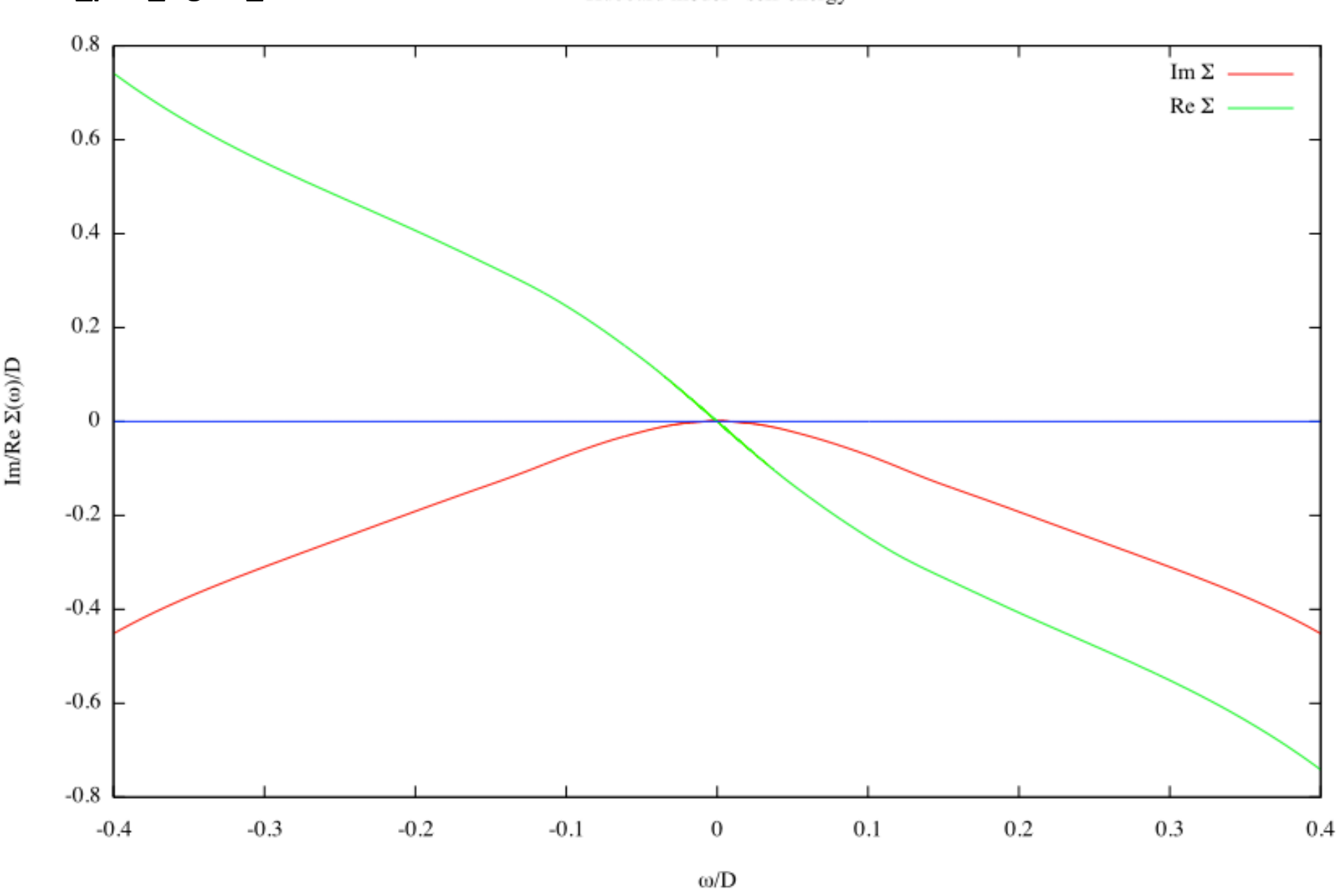
46_Hubbard_MIT



3b_plot_sigma



3b_plot_sigma_zoom



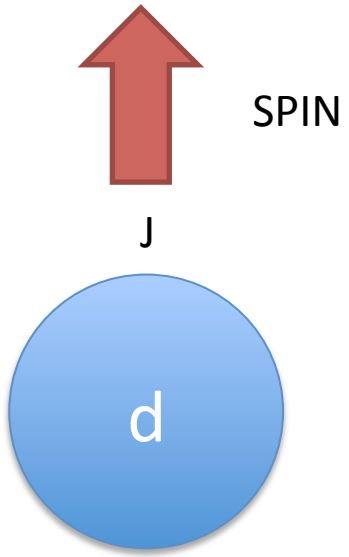
Exercises

- Increase U and locate the MIT at $U=U_{c2}$.
- Starting from an insulating initial approximation, reduce U and locate the MIT at $U=U_{c1}$.
- Do an U-sweep at finite temperature T=0.06D. Observe the cross-over from metal-like to insulator-like spectral functions.

Kondo lattice model

$$H = \sum_{k\sigma} \epsilon_k c^\dagger_{k\sigma} c_{k\sigma} + \sum_i J \mathbf{s}_i \cdot \mathbf{S}_i$$

```
def1ch[1];

If[!paramexists["spin", "extra"],
   MyError["Define the spin of the impurity!"];
   ];
SPIN = ToExpression @ param["spin", "extra"];
MyPrint["SPIN=", SPIN];

Module[{sz, sp, sm, sx, sy, oz, op, om, ss},
       sz = spinketbraZ[SPIN];
       sp = spinketbraP[SPIN];
       sm = spinketbraM[SPIN];
       sx = spinketbraX[SPIN];
       sy = spinketbraY[SPIN];

       oz = nc[ sz, spinz[ d[] ] ];
       op = nc[ sp, spinminus[ d[] ] ];
       om = nc[ sm, spinplus[ d[] ] ];

       ss = oz + 1/2 (op + om) // Expand;

       Hk = Jkondo ss;
       ];
```



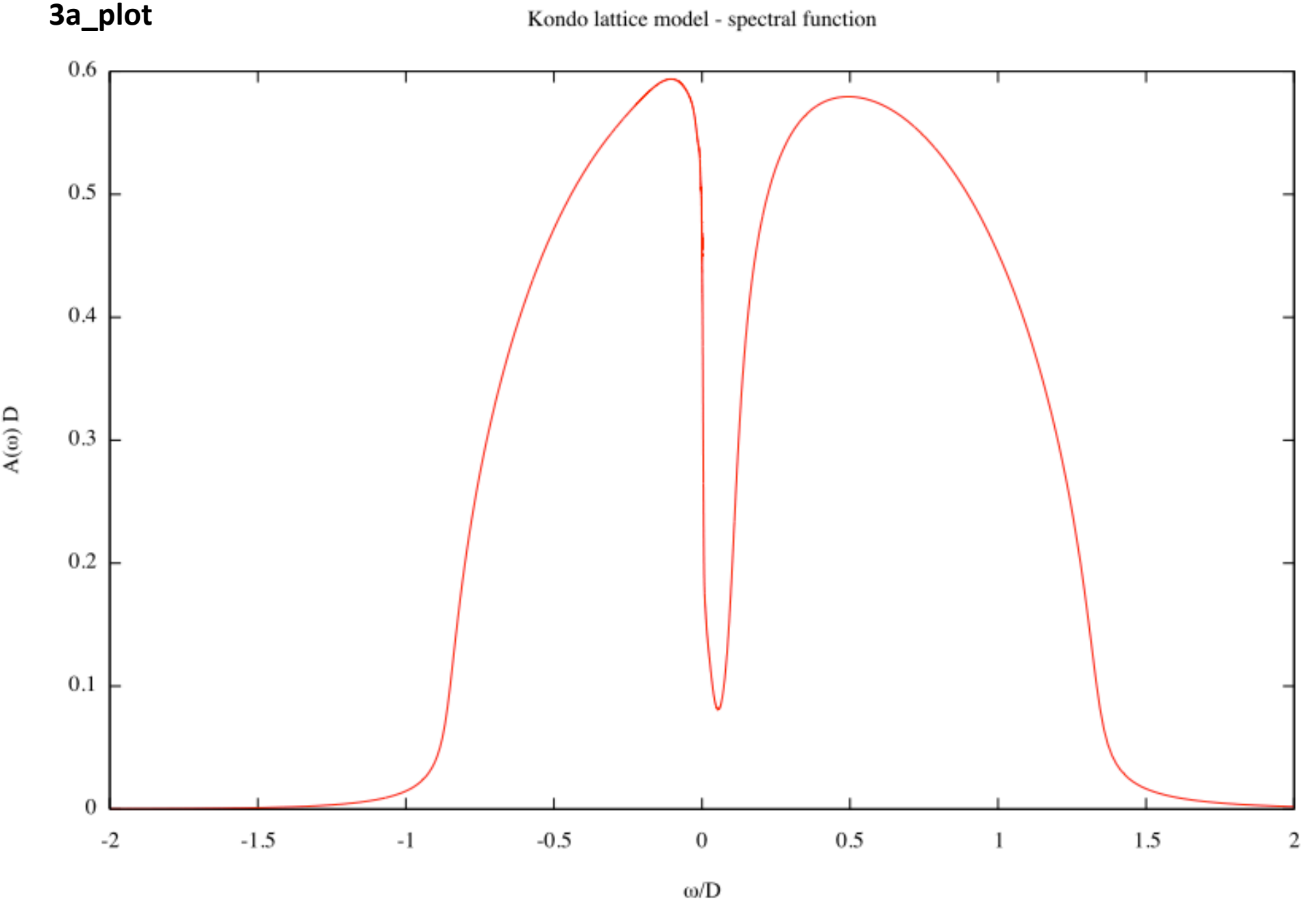
model.m

```
H = H0 + H1 + Hc + Hk;

MAKESPINKET = SPIN;

(* All operators which contain d[], except hybridization (Hc). *)
Hselfd = H1 + Hk;

selfopd = ( Chop @ Expand @ komutator[Hselfd /. params, d[#1, #2]] )&;
```

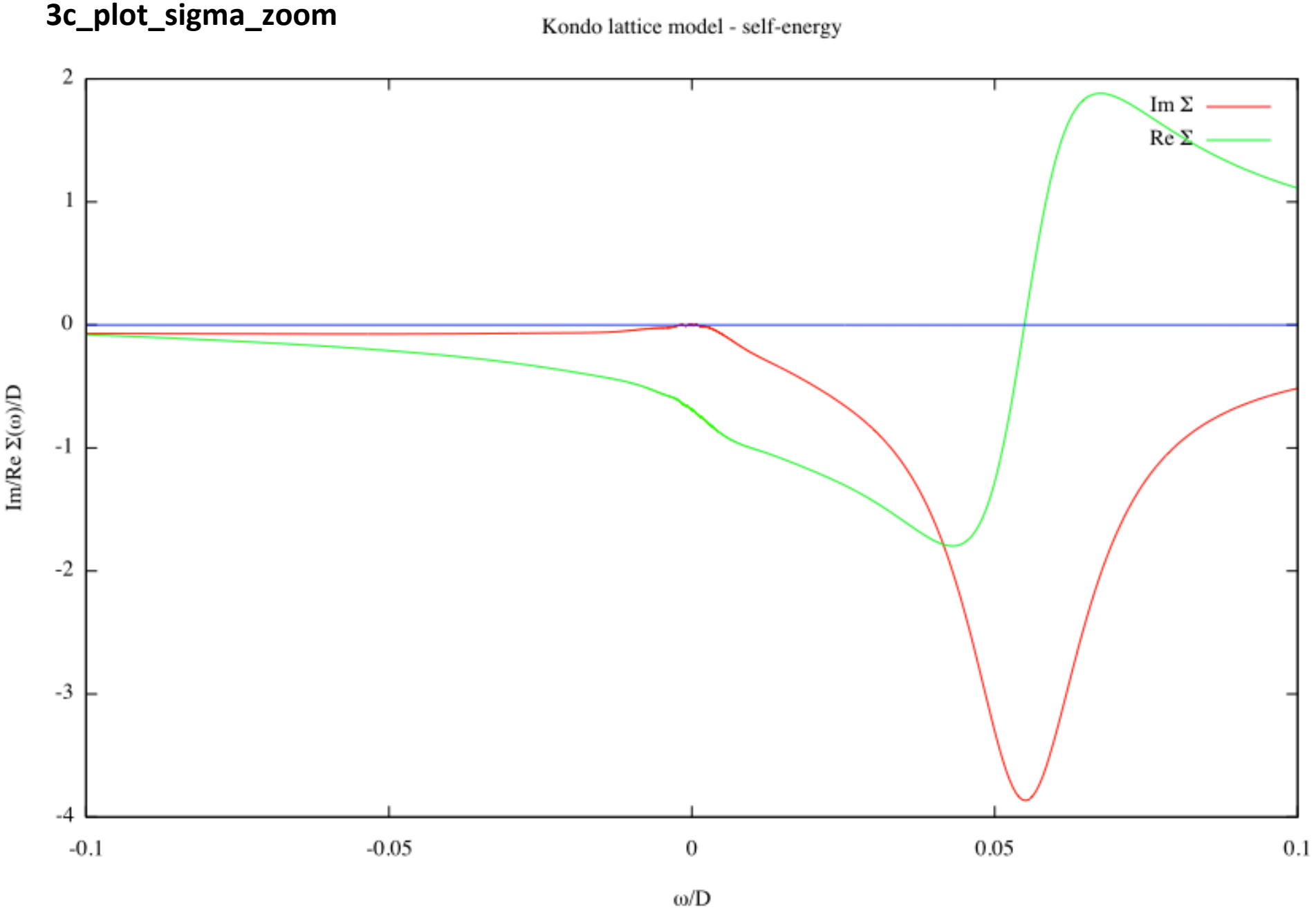
```
[extra]
spin=1/2
Jkondo=0.05

[param]
xmax=15
adapt=false

tri=cpp
preccpp=2000
prec=100

symtype=QS
model=../model.m

fixeps=1e-10

U=0
delta=$DELTA
Gamma=-999
```

param.loop

3a_plot


Kondo lattice model - spectral function
A(ω) D
ω/D
0.6
0.5
0.4
0.3
0.2
0.1
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

3c_plot_sigma_zoom

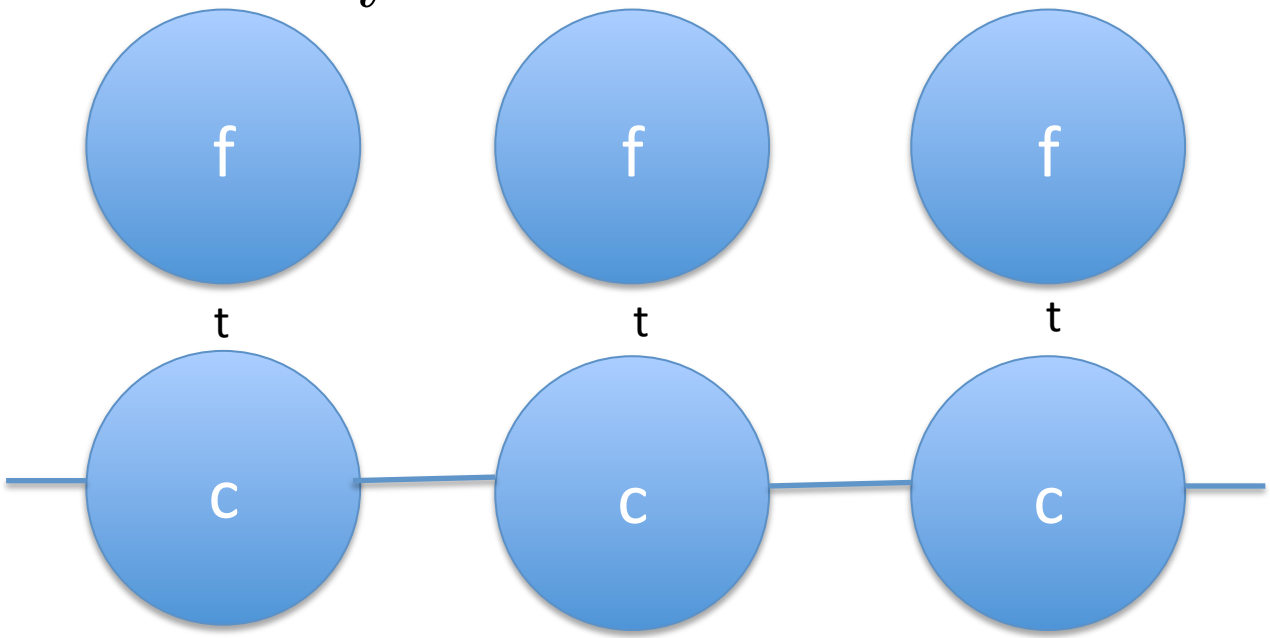


Exercises

- Change the target occupancy (parameter goal in param.loop) towards 1. What happens?
- Increase J. How does the width of the pseudogap decrease?

Periodic Anderson model

$$H = \sum_{k\sigma} \epsilon_k c^\dagger_{k\sigma} c_{k\sigma} + \sum_{i\sigma} \left(\epsilon_f f^\dagger_{i\sigma} f_{i\sigma} + U_f n_{i\uparrow} n_{i\downarrow} \right) + \sum_i t \left(f^\dagger_{i\sigma} c_{i\sigma} + \text{H.c.} \right)$$

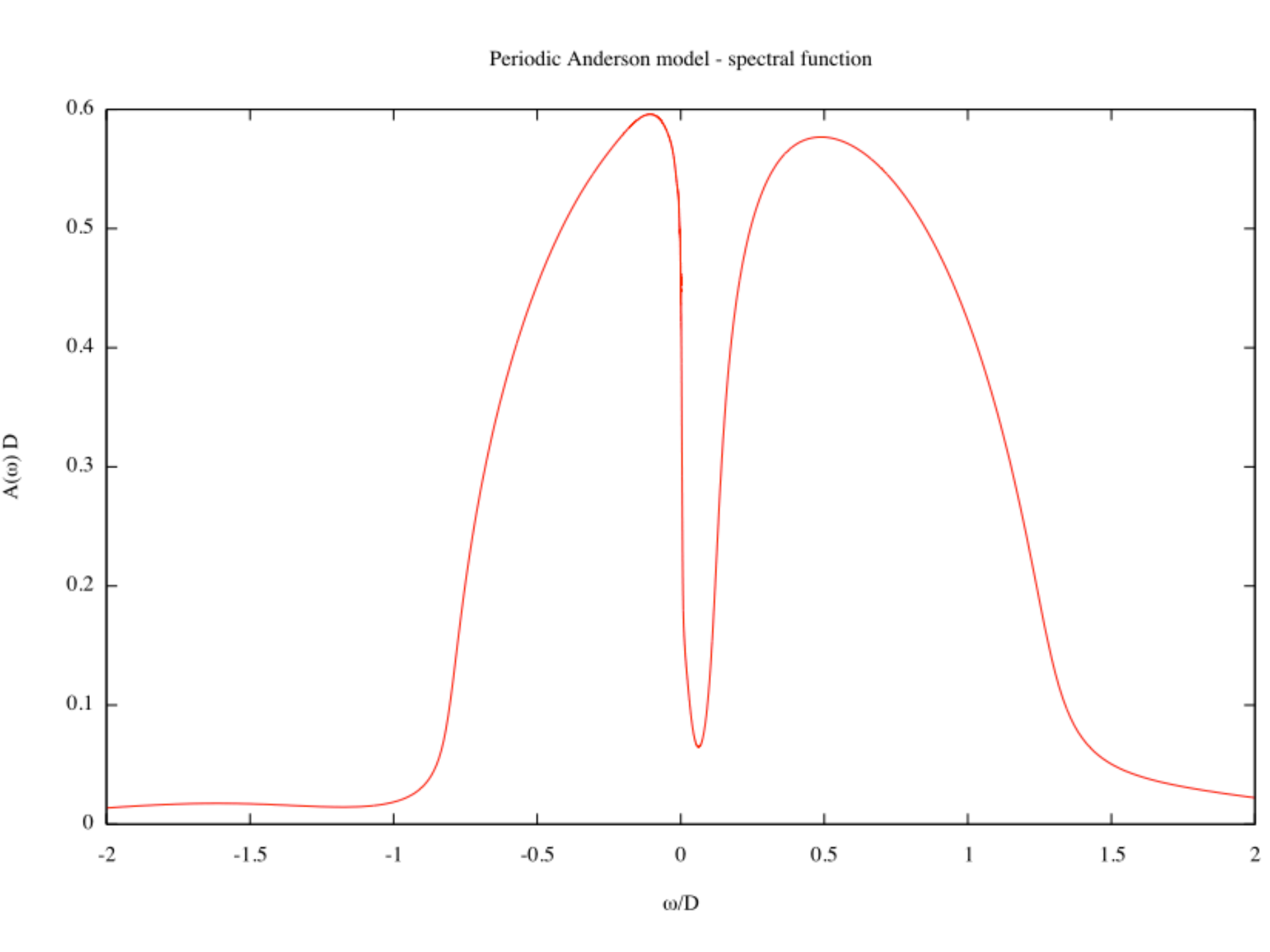


```
def1ch[2];

Ha = eps number[a[]] + Uf hubbard[a[]];
H1a = t hop[d[], a[]];

H = H0 + H1 + Hc + Ha + H1a;

(* All operators which contain d[], except hybridization (Hc). *)
Hselfd = H1 + H1a;

selfopd = ( Chop @ Expand @ komutator[Hselfd /. params, d[#1, #2]] )&;
```

model.m

```
[extra]
Uf=0.3
eps=-0.15

[param]
xmax=15
adapt=false

tri=cpp
preccpp=2000
prec=100

symtype=QS
model=../model.m

fixeps=1e-10

U=0
t=0.05
delta=$DELTA
Gamma=-999
```

param.loop


Periodic Anderson model - spectral function
A(ω) D
ω/D
0.6
0.5
0.4
0.3
0.2
0.1
0
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Exercises

- Decrease U_f towards 0. What happens?
- Increase t. What happens?