JointInversion_Overview

March 25, 2020

1 Joint Inversion Tutorial

General Layout of Document:

1.1 Setup

1.2 Vs Starting Model (STA_data/mod.STA) & search region (MonteCarlo/in.para_STA)

1.3 Joint inversion parameters to edit within C-codes

1.4 Joint inversion edits within csh code & general setup

1.5 Run joint inversion & General output

1.6 Plotting joint inversion results

2 Setup

2.1 Data files : STA_data (example: I23K_data)

Scale uncertainties that go into these files as needed before running the joint inversion

2.1.1 Phase Velocity

File: STA_data/STA.ph Columns: Period Vph Uncertainty

2.1.2 H/V

File: STA_data/STA.HV Columns: Period H/V Uncertainty

2.1.3 Receiver functions

File: STA_data/STA.RF Columns: Time(s) RF_Amplitude Uncertainty

*Note that if you do **not** have receiver functions, you will set the joint inversion to use **MonteCarlo/in.rf** within the **MonteCarlo/do_one.csh file f4** parameter (see comments within this file)*

Notes:

A) You may need to ***scale*** the amplitude of provided receiver functions.

Here I am using a ***gaussian width of 3.0 and a ray parameter of 0.06***, but I scale the amplitudes of the receiver functions as: ***JointInversionRF_input=RFamp/1.7***

Scaling information from: http://eqseis.geosc.psu.edu/cammon/HTML/RftnDocs/seq01.html (final table)

Gaussian Width Factor - Divide Iterdecon Rftn By

0.5 - 0.29

1.0 - 0.57

1.5 - 0.85

2.5 - 1.42

3.0 - 1.70

5.0 - 2.83

B) Be sure to include 0.00s time (I linearly interpolate back to estimate this point if not provided)

C) All receiver functions should have the same sampling rate, and the memory allocation size is currently set up to 0.05s sampling rate from 0 to 10s. The RF data allowance in the joint inversion is set:

1. **maxdata = 1024**

set in **RF/rfi_param.inc**

Note that this should be a factor of 2 (so 2^X where X is a positive integer)

If you have more data points (higher sampling rate or longer receiver function), you may need to increase this.

D) Check that the receiver functions are set correctly in **CALforward.C** :

1. Gaussian width **(float gau = 2.5)**
2. Ray parameter **(float slow=0.06)**

---

3 Vs Starting Model (STA_data/mod.STA) & search region (MonteCarlo/in.para_STA)

Nearest point to a given station's location is used to create a starting model, including Moho (crustal thickness) and top layer sediment thickness estimates. From this information, the parameter search range for the crustal thickness and sediment layer is set.

3.1 Python file MCMC_SetupData/setup.py

Python file to find starting model & set up STA_data/mod.STA and MonteCarlo/in.para_STA:

MCMC_SetupData/setup.py

Change as needed in **lines 36-116 (settings)**, but should not need to change much after that. See: Sediment / Crust / Mantle mod.STA settings!! (lines- settings: 66-74; application: 438-470) Also, see: in.para_STA settings (lines settings: 53-60; application: 60-78, 497-502)

Run via: python setup.py

3.2 Details of STA_data/mod.STA (used in setup.py)

Format & Example of STA_data/mod.STA : (again, **generated in python script for a given lat/lon of a station,** but please check the output is formatted how you would like)

0 4 4.0 2 1.197000 3.483000 1 2 1 0 0 0 0 0 0 0 0

1 2 28.0 4 3.471627 3.681347 3.464843 3.819802 1 3 1 0 0 0 0 0 0 0 0

2 2 112.0 5 3.953653 4.380285 4.320212 4.376671 4.376664 2 4 2 0 0 0 0 0 0 0 0

Explanation of Columns:

Column 1: ids of the layer (0,1,2)

COlumn 2: parameterization type flag: (4: linear; 2: cubic B-splines; 1: Layerized)

Column 3: thickness of each layer (km)

Column 4: number of values that describe the layer (2 for linear, N for B-spline, N for layerized)

Column 5-5+N(2): values that describe the layer

Column -11: Rflag, 1(Brocher's 2005 paper); 2(Perturbation relative to reference velocity – Kaban, M. K et al. (2003), Density of the continental roots [use for mantle]); 3(1.05, for water)

Column -10: Qflag, 0(input), 1(water: 0/57882); 2(sediment,160/80); 3(crust,600/1368), 4(mantle,417/1008)

Column -9: Vp flag, (see CALgroup.C get_Vp): 1 (Brocher, 2005); 2 (AK135, 120km, vpvs=1.789); 3 vpvs = vpvs [where vpvs is set at Column -8 and vpvs1 is set at Column -7]; 4 vp = vs*(1.373+2.022/vs)

CALmodel.C also has details on each column, but I do not use the last 8 columns, so am unsure on if the code handles setting these to anything other than 0 correctly.

3.3 Details of MonteCarlo/in.para_STA (implemented in setup.py)

Format & Example of MonteCarlo/in.para_STA :

(also **set up in python script**, but please check that this is formatted how you need and change the python script to output new ranges, etc, as needed)

```
0 1 1.5 0.05 0 0
0 1 1.5 0.05 0 1
0 -1 20 0.05 1 0
0 -1 20 0.05 1 1
0 -1 20 0.05 1 2
0 -1 20 0.05 1 3
0 -1 20 0.05 2 0
0 -1 20 0.05 2 1
0 -1 20 0.05 2 2
0 -1 20 0.05 2 3
0 -1 20 0.05 2 4
1 1 100 0.1 0
1 1 5.6 1.0 1
```

Column Information:

Column 1: id_flag: 0(values to be perturbed), 1(thickness to be perturbed)

Column 2: pertur_flag 1(absolute perturbation), -1(relative perturbation in percentage)

Column 3: perturbation range

Column 4: gaussian step width

Column 5: layer id

Column 6: value id

Row Information:

Rows 1&2: Sediment layer Vs perturbation range (top & bottom of linear layer)

Rows 3-6: Crustal layer Vs perturbation range (B-spline values used to create Vs curve)

Rows 7-11: Manle layer Vs perturbation range (B-spline values)

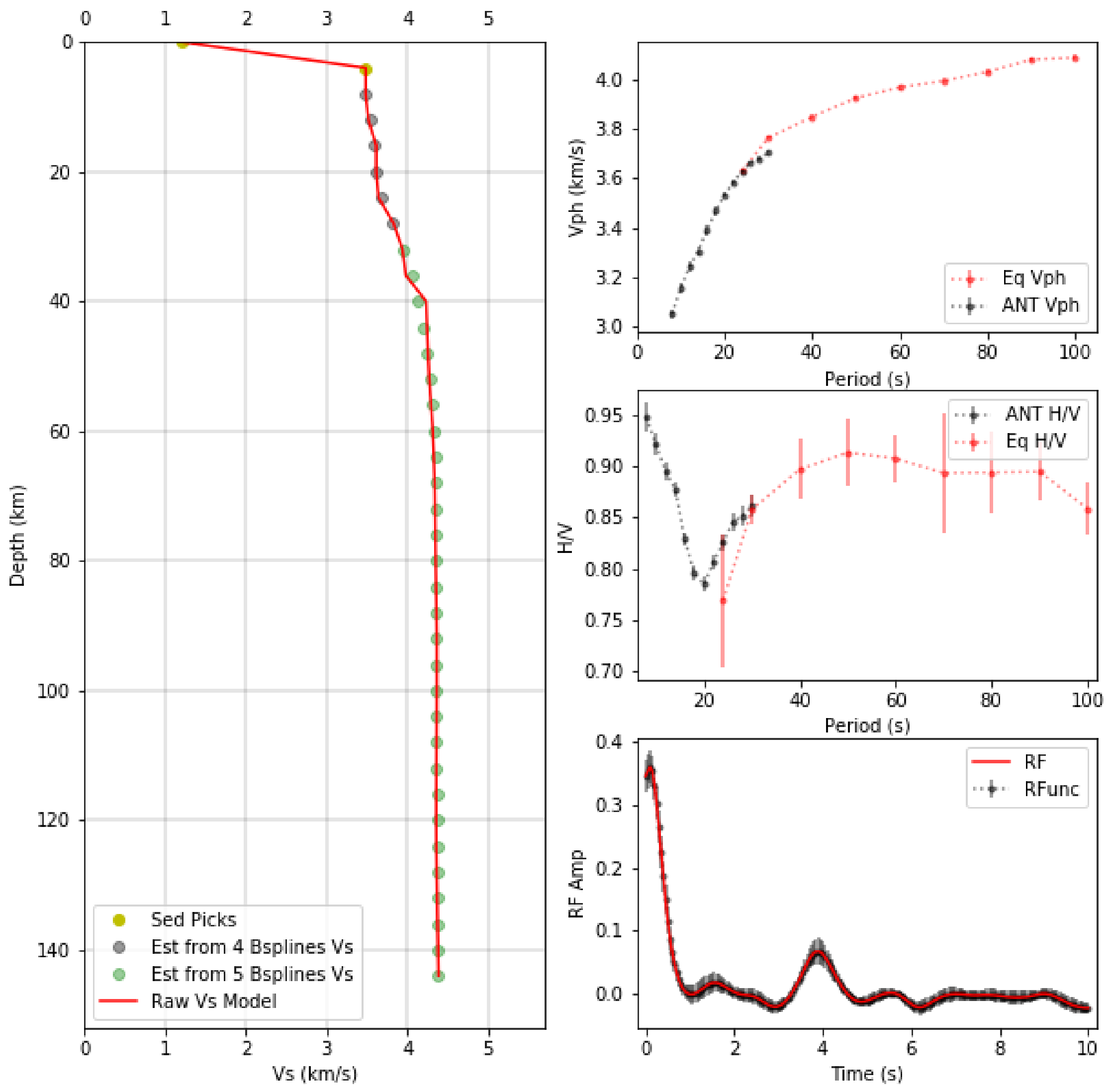
Row 12: Sediment thickness

Row 13: Crustal thickness

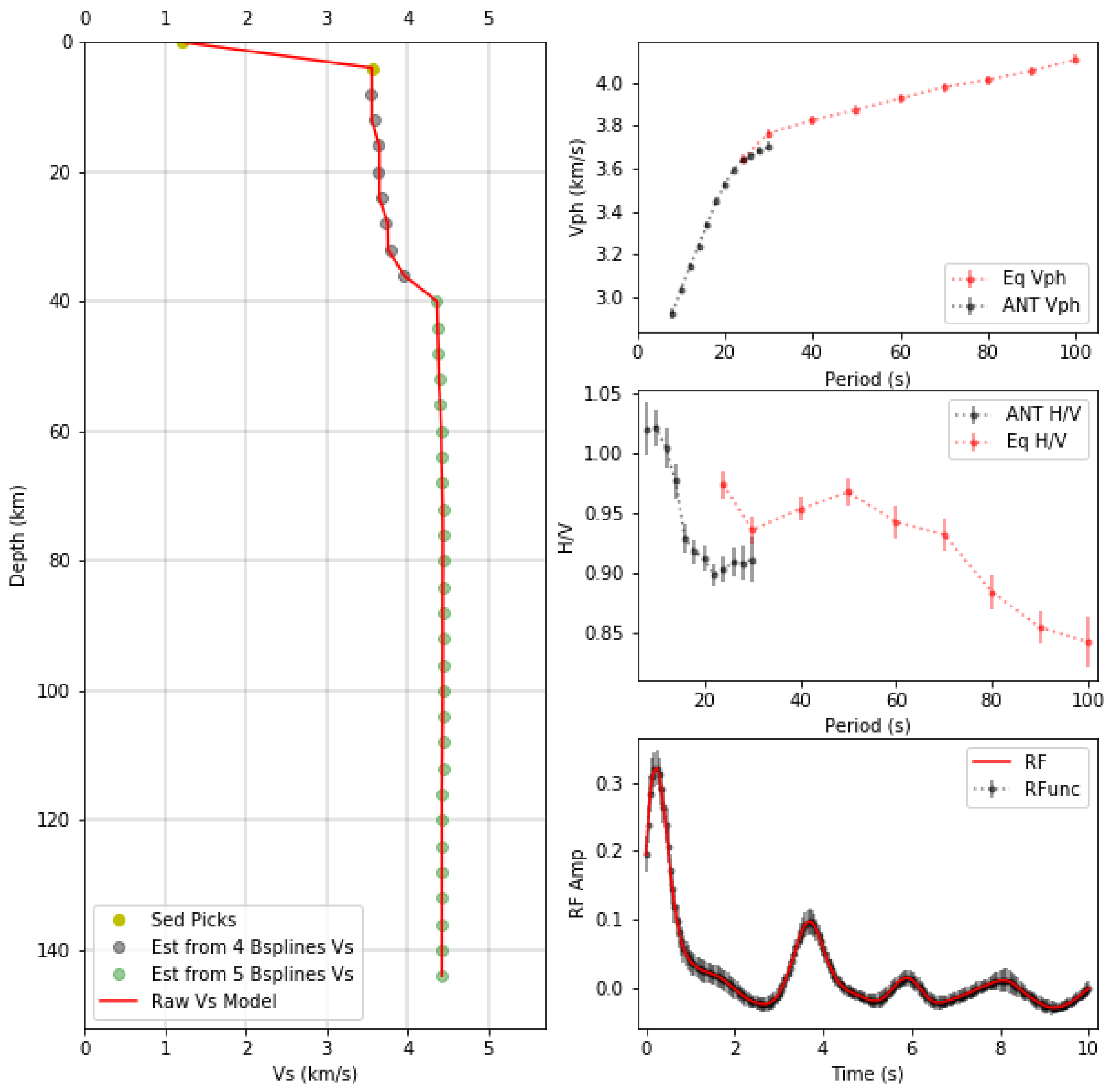
Note that I tend to perturb crustal thickness by absolute value of km as this seems to be more stable & easier to check the range is reasonable (you can also set minimum search or maximum search range as needed)

[56]:
```python
# Example of output for starting model for station CAST (see CAST_data for␣
↪details)
# Run setup.py
#   to generate the CAST_data/mod.CAST, MonteCarlo/in.para_CAST,
#   and query_dir/StartingModel_CAST.png files (files also created for I23K)
#   references MCMC_TestData/stations.lst and
#   requires STA_data to already exist with .HV, .ph, and .RF files included
#   run from terminal using python3: python setup.py

#from IPython.display import Image
from PIL import Image

pltdir='/uufs/chpc.utah.edu/common/home/u1015716/JointInversion_Tutorial/
↪MCMC_TestData/query_dir'

imgsize=[500,500]

startmod=pltdir+'/StartingModel_CAST.png'
f=Image.open(startmod)
print('\n   Starting model and Data for CAST station (generated via␣
↪MCMC_SetupData/setup.py) :')
display(f)
print('\n\n\n\n')

startmod=pltdir+'/StartingModel_I23K.png'
f=Image.open(startmod)
print('\n   Starting model and Data for I23K station (generated via␣
↪MCMC_SetupData/setup.py) :')
display(f)
print('\n\n\n\n')
```

```
   Starting model and Data for CAST station (generated via
MCMC_SetupData/setup.py) :
```



Starting model and Data for I23K station (generated via MCMC_SetupData/setup.py) :



4 Joint inversion parameters to edit within C-codes

Note that to recompile the C-codes, use:

make clean

make do_MC_Para

make do_MC_Para_Post_process_v3

4.1 CALforward.C - Receiver function parameters and space

As previously discussed, make sure that the receiver function allocation is adequate, and that the gaussian width and ray parameter are set correctly.

From above

C) All receiver functions should have the same sampling rate, and the memory allocation size is currently set up to 0.05s sampling rate from 0 to 10s. The RF data allowance in the joint inversion is set:

1. **maxdata = 1024**

set in **RF/rfi_param.inc**

Use 'make clean' followed by 'make' in the terminal to recompile if this parameter is changed

If you have more data points (higher sampling rate or longer receiver function), you may need to increase this.

D) Check that the receiver functions are set correctly in **CALforward.C** :

1. Gaussian width **(float gau = 2.5)**
2. Ray parameter **(float slow=0.06)**

4.2 CALgroup.C - get_vp & get_rho

Can edit the equations used here as needed to calculate vp and density. I would suggest hard-coding specific Vp/Vs ratios or density calculations and reference them here from the set flag (mod.STA – column-9 for Vp)

4.3 CALmodel.C - (A) weights between provided datasets & (B) model requirements (goodmodel)

4.3.1 A) Weights between provided datasets - see: int compute_misfit

Decide what weights you would like between the datasets.

Weighted Example: For example, if want to downweight phase velocity, but keep H/V and receiver function weights high, you might want misfit and likelihood to have:

20% contribution from Vph, 40% from H/V, and 40% from RF

(or, if no receiver function, then 66% H/V and 33% phase velocity).

In this case, the **phase velocity and H/V dispersion misfit** needs to be changed:

1. model.data.disp.misfit=(0.667)*tmfe+(0.333)*tmfp;

Note that tmfe=misfit from H/V; tmfp=misfit from phase velocity

Also note that the S1 is not changed here, as the overall tS is specified later on and including here could result in problems with forgetting to change it back

And the **final combined misfit & likelihood** needs to be changed (most important):

2. tS = 0.5*((0.4*Se)+(0.2*Sp)+(0.4*S2));
3. model.data.misfit=(0.2*tmfp)+(0.4*tmfe)+(0.4*tmisfit2);

Non-weighted Example: In this case, the **phase velocity and H/V dispersion misfit** should be set as:

1. model.data.disp.misfit=tmisfit1;

Note: tmisfit1=sqrt((tempv1+tempv3)/(model.data.disp.npper+model.data.disp.neper)); //original, no group velocity

And the **final combined misfit & likelihood** needs to be changed (most important):

2. tS = 0.5*(inp*S1+(1-inp)*S2);

Note: for joint inversion, inp should be 0.5; for RF only, inp == 0; for disp-only, inp == 1.

3. model.data.misfit=inp*tmisfit1+(1-inp)*tmisfit2;;

4.3.2 B) model requirements (goodmodel)

The CALmodel.C file should be relatively self-explanatory, as each section is begins and ends with:

// *** //

where you need to only add/remove to include :

1. /* at the beginning of a section and
2. */ at the end of a section

Extra commented out lines are available as references for options within each section.

Optional criteria for models includes:

1. checks model layers<4.9 in crust and sed layers (e.g., non-mantle)
2. check bottom of overlying layer Vs (e.g., crust) < top of deeper layer (e.g., mantle) Vs
3. check 1st crust spline < 2nd crust spline
4. check monotonically increasing (haven't used this one in a long time, so use with caution)
5. Check sediment layer is increasing Vs with depth (if not, discard model)
6. checking for postitive gradient at top of layers

4.4 do_MC_Para.C & do_MC_Para_Post_process_v3.C

4.4.1 do_MC_Para.C

Purpose: Read in data and starting model. Perform forward model of the starting model (output to file Initial.hv, Initial.rf, and Initial.ph). Create files to store information, prepare to perform in parallel as needed, and perform inversion one jump at a time (one jump set per core – see MC.C for details). Write everything to file.

```
No changes to this file should be necessary.
```

4.4.2 do_MC_Para_Post_process_v3.C

Purpose: Go through all models that are generated in the inversion (and pass the model criteria set in goodmodel) and determine min/max of searched range in Vs, which models consist of the posterior, the average of the posterior, the minimum misfit model, and forward modeling phase velocity, receiver function, and H/V expected for the posterior models (and final minimum misfit and final average of posterior models).

Optional edits:

1. int embnlays=151;

   ```
   This sets the maximum depth to search for vmin and vmax.
   Edit as needed for your data (embnlays=151 corresponds to 0-150km depth).
   The search range is also set to increase by 1km steps with depth (see temp_demb).
   ```

2. Go through and output information for every model (not just posterior)

   ```
   This long block of code is commented out, but you can add it back in if you want
   (will take forever to run and output info to file though,
   so don't do this unless using a small number of jumps and iterations).
   ```

3. Minimum and maximum of all parameters are also output, so if you want to try and see how much each specific parameter searches, you can do that (outputs to *.minparas and *.maxparas).

4. Output posterior parameters (*.all.parameters), Vs (*.all.value), depth (*.all.value_depth), Vp (*.all.value_vp), density (*.all.value_rho) to file.

5. Output forward model results for posterior models' H/V (*.all.e), receiver functions (*.all.rf), phase velocities (*.all.ph), and group velocities (*.all.gr).

   ```
   Note that you can comment these out if you don't need them
   (e.g., group velocity is commented out).
   ```

---

5 Joint inversion edits within csh code & general setup

Include the following files into the MonteCarlo directory:

1. MonteCarlo/in.rf (use if no receiver function available at this station)

2. MonteCarlo/Qmodel.dat

5.1 See MCMC_TestData/MonteCarlo/do_one.csh for details on edits

5.1.1 Receiver function places to change:

ff4=

gw= (ALSO be sure this is set correctly in CALforward.C,

along with ray parameter [slow])

echo p XX » $fcontrol

(XX=1 for only dispersion [H/V, phase velocity, group velocity];

XX=0.5 receiver function and dispersion information used)

5.1.2 H/V, phase velocity, and group velocity (dispersion measurements) places to change:

```
    echo disp X Y 1 $ff1 2 $ff5 3 $ff2

where this is set up as follows:

X=1 for Rayleigh; Y=Number of inputs; 1 phasevel_filename;
2 groupvel_filename; 3 H/V_filename;
(can disclude files if not available, and change number of inputs (Y) accordingly)

Examples:

#echo disp 1 1 1 $ff1 >> $fcontrol
# 1=Rayleigh,1=number of inputs [justphase(no H/V)],1 --> phase designated file

echo disp 1 2 1 $ff1 3 $ff2 >> $fcontrol
# 1=Rayleigh,2=number of inputs [phase&H/V],
# 1 --> phase file, 3 --> h/vfile

#echo disp 1 3 1 $ff1 2 $ff5 3 $ff2 >> $fcontrol
# 1=Rayleigh,3=number of inputs [phase&group&H/V],
# 1 --> phasefile, 2--> group vel, 3 --> h/v file
```

```
echo p XX >> $fcontrol

(XX=1 for only dispersion [H/V, phase velocity, group velocity];
XX=0.5 receiver function and dispersion information used)
```

5.1.3 Number of jumps

echo tt 12 » $fcontrol # 12 jumps

6 Run joint inversion & general output

Once the settings are how you prefer, then run the inversion.

Here are a couple of different options pre-created for you to try and use, and as reference for changing the codes as needed in the future.

6.1 Preset Code Options:

Note that as forward modeling occurs over all posterior models (see do_MC_Para_Post_Process_v3.C) for H/V, receiver function, and phase velocity data are forward modeled even if not used in the inversion (you can comment this out as you wish in the code). Also, all are plotted because of how the python script is set up.

After edits to C-codes, re-make executables via:

```
make clean
make do_MC_Para
make do_MC_Para_Post_process_v3
```

Run via (for example 1 and station I23K):

```
cd MCMC_TestData
sh MonteCarlo/do_one_wtRFVphHV__4pt9kms_posCrustToMantle.sh I23K 1000
    > output_I23K_tmp.txt

mv output_I23K\*txt I23K/.
python ../MCMC_AnalyzeResult/inversion_plot_vfinal.py I23K
    JointInversion_Tutorial/MCMC_TestData

mv I23K I23K_wtRFVphHV__4pt9kms_posCrustToMantle
```

Note that output_I23K_tmp.txt and output_I23K.txt are used to look at the number of iterations vs. misfit and misfit required for a model to be in the posterior.

Also, this is a relatively short run to provide an example. ***I usually have 12 jumps (echo tt 12 [see MonteCarlo/do_one.sh]) and 3000 (instead of 1000) iterations.***

A good rule of thumb is that the posterior should consist of 1000 or so models, and that ensures you're sampling the posterior effectively. You may need to increase the breadth of parameter model/parameter space searched, and the number of jumps and number of iterations per jump.

6.1.1 Weighted joint inversion of phase velocity, H/V, and receiver function (3.0-gaussian width, 0.05s sampling rate, and 0.06 ray parameter). Enforce model layers < 4.9km/s in crust&sed layer, positive jump from crust into mantle

```
MCMC Directory: MCMC_Codes/MCMC_RFgw3_weightmisfit_gt4pt9_posCrustToMantle

MonteCarlo csh Code: MCMC_Codes/MonteCarlo/
    do_one_wtRFVphHV__4pt9kms_posCrustToMantle.sh
```

```
Output directory moved to: CAST_wtRFVphHV__4pt9kms_posCrustToMantle

C-Code Edits:
    1. CALforward.C (slow=0.06; gau=3.0)

    2. CALgroup.C & setup.py -->
        Brocher for Vp & density in sed & crust (flag==1)
        Vp/Vs=1.789, 10km/m3 per1% in mantle (flag==2)

    3. CALmodel.C - int compute_misfit
        model.data.disp.misfit=(0.667)*tmfe+(0.333)*tmfp;
        tS = 0.5*((0.4*Se)+(0.2*Sp)+(0.4*S2));
        model.data.misfit=(0.2*tmfp)+(0.4*tmfe)+(0.4*tmisfit2);

    4. CALmodel.C - int goodmodel
        Use 'checks model layers<4.9'
        Use 'check bottom of overlying layer Vs'
            (i=1; i<model.ngroup-1; i++)//between crust and mantle
        Do not use other model checks

 do_one*csh edits:
    1. ff4=$mdir"/"${sta}"_data"/${sta}.RF
    2. gw=3.0
    3. echo p 0.5
    4. dircode path
```

6.1.2 Weighted joint inversion of phase velocity & H/V (receiver function settings set to be applicable for in.rf - 2.5 gaussian width, 0.06 ray parameter, and 0.1s sampling rate). Enforce model layers $<$ 4.9km/s in crust&sed layer, positive jump from crust into mantle, and check 1st mantle spline value $<$ 2nd mantle spline value

```
MCMC Directory:
    MCMC_RFgw3_weightmisfitDispOnly__gt4pt9_posCrustToMantle_1MsplineLT2Mspline

MonteCarlo csh Code:
    MonteCarlo/do_one_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline.sh

Output directory moved to:
    I23K_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline

C-Code edits:
    1. CALforward.C (slow=0.06; gau=2.5)

    2. CALgroup.C & setup.py --> no changes
    3. CALmodel.C - int compute_misfit
        model.data.disp.misfit=(0.667)*tmfe+(0.333)*tmfp;
        tS = 0.5*((0.67*Se)+(0.33*Sp));
        model.data.misfit=(0.33*tmfp)+(0.67*tmfe);

    4. CALmodel.C - int goodmodel
        Use 'checks model layers<4.9'
        Use 'check bottom of overlying layer Vs'
            (i=1; i<model.ngroup-1; i++)//between crust and mantle
        Use //check 1st crust spline < 2nd crust spline

do_one*csh edits:
    1. ff4=$mdir"/MonteCarlo/in.rf"
    2. gw=2.5
    3. echo p 1
    4. dircode path
```

6.1.3 Non-weighted joint inversion of phase velocity, H/V, and receiver function (3gaussian width, 0.06 ray parameter, 0.05s sampling rate). Enforce model layers < 4.9km/s in crust&sed layer, and positive jump from crust into mantle

```
MCMC Directory: MCMC_Codes/MCMC_RFgw3__gt4pt9_posCrustToMantle

MonteCarlo csh Code: MonteCarlo/do_one_RFVphHV__4pt9kms_posCrustToMantle.sh

Output directory moved to: I23K_RFVphHV__4pt9kms_posCrustToMantle

C-Code Edits:
    1. CALforward.C (slow=0.06; gau=3.0)

    2. CALgroup.C & setup.py --> no change

    3. CALmodel.C - int compute_misfit
        model.data.disp.misfit=tmisfit1;
        tS = 0.5*(inp*S1+(1-inp)*S2);
        model.data.misfit=inp*tmisfit1+(1-inp)*tmisfit2;

    4. CALmodel.C - int goodmodel
        Use 'checks model layers<4.9'
        Use 'check bottom of overlying layer Vs'
            (i=1; i<model.ngroup-1; i++)//between crust and mantle
        Do not use other model checks

 do_one*csh edits:
    1. ff4=$mdir"/"${sta}"_data"/${sta}.RF
    2. gw=3.0
    3. echo p 0.5
    4. dircode path
```

6.1.4 Non-weighted joint inversion of phase velocity & receiver function (3gaussian width, 0.06 ray parameter, 0.05s sampling rate). Enforce model layers < 4.9km/s in crust & sed layers, positive jump between crust into mantle, positive jump from sediment to crust, and sediment layer increasing

Note that this still requires an H/V data input, but the H/V datafile used has extremely high uncertainty and therefore doesn't impact the inversion

```
awk '{print $1,$2,500}' I23K.HV > I23K.HV_nounc

MCMC Directory: MCMC_Codes/MCMC_RFgw3__gt4pt9_posCrustToMantle_SedIncreasing

MonteCarlo csh Code:
    MonteCarlo/do_one_RFVph__4pt9kms_posCrustToMantle_SedIncreasing.sh

Output directory moved to: I23K_RFVph__4pt9kms_posCrustToMantle_SedIncreasing

C-Code Edits:
    1. CALforward.C (slow=0.06; gau=3.0)

    2. CALgroup.C & setup.py --> no change

    3. CALmodel.C - int compute_misfit
        model.data.disp.misfit=tmisfit1;
        tS = 0.5*(inp*S1+(1-inp)*S2);
        model.data.misfit=inp*tmisfit1+(1-inp)*tmisfit2;

    4. CALmodel.C - int goodmodel
        Use 'checks model layers<4.9'
        Use 'check bottom of overlying layer Vs'
            (i=1; i<model.ngroup-1; i++)//between crust and mantle
        Use 'Check sediment layer is increasing Vs with depth'

 do_one*csh edits:
    1. ff4=$mdir"/"${sta}"_data"/${sta}.RF
    2. ff2=$mdir"/"${sta}"_data"/${sta}.HV_nounc
    3. gw=3.0
    4. echo disp 1 2 1 $ff1 3 $ff2 >> $fcontrol
        Note: use echo disp 1 1 1 $ff1 >> $fcontrol
        if don't want to consider H/V at all
        (and remove H/V loading information from python plotting script)
    5. echo p 0.5
    6. dircode path
```

7 Plotting joint inversion results

See **JointInversion_Tutorial/MCMC_AnalyzeResult/inversion_plot_vfinal.py** to plot.

Run via:

python MCMC_AnalyzeResult/inversion_plot_vfinal.py I23K JointInversion_Tutorial/MCMC_TestData

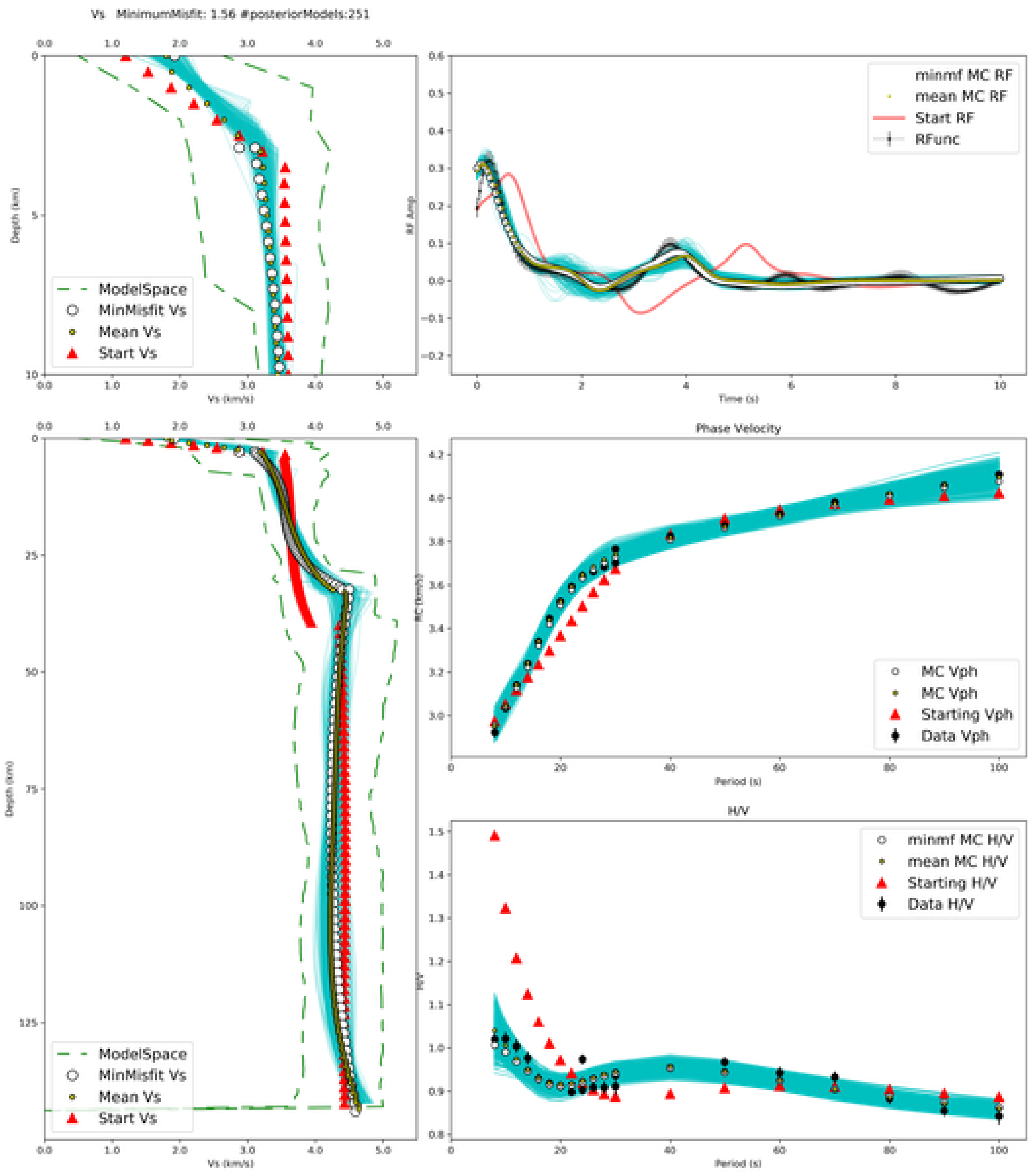
or

python MCMC_AnalyzeResult/inversion_plot_vfinal.py CAST JointInversion_Tutorial/MCMC_TestData

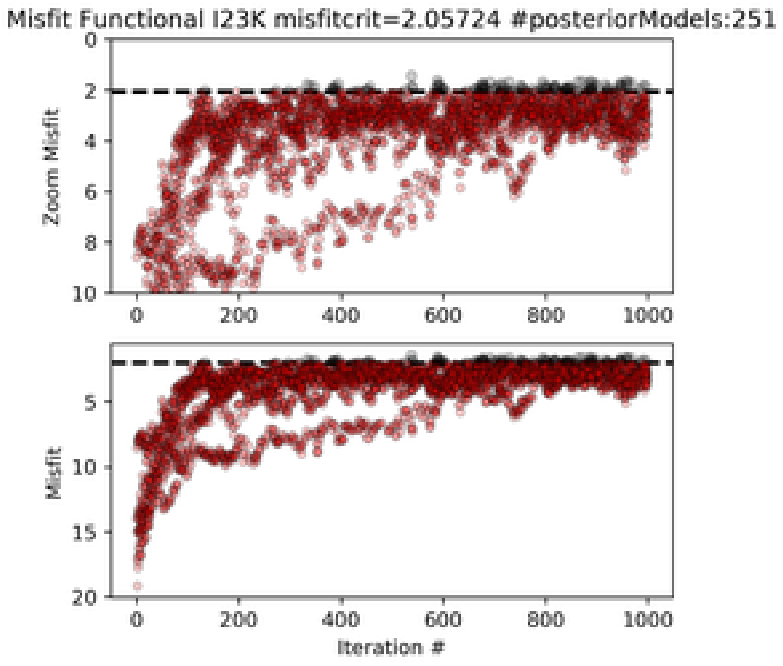
[62]:
```
# Example of output for starting model for station CAST (see CAST_data for␣
 ↪details)
# Run via (for example 1 and station I23K):
# cd MCMC_TestData
# sh MonteCarlo/do_one_wtRFVphHV__4pt9kms_posCrustToMantle.sh I23K 1000 >␣
 ↪output_I23K_tmp.txt
# mv output_I23K*txt I23K/.
# python ../MCMC_AnalyzeResult/inversion_plot_vfinal.py I23K␣
 ↪JointInversion_Tutorial/MCMC_TestData
# mv I23K I23K_wtRFVphHV__4pt9kms_posCrustToMantle



pltdir='/uufs/chpc.utah.edu/common/home/u1015716/JointInversion_Tutorial/
 ↪MCMC_TestData'

imgsize=[600,600]
imgsize1=[300,300]


#####################################################
# from Preset Code Option 1 #

print('Joint inversion, weighted datasets')

newf=pltdir+'/I23K_wtRFVphHV__4pt9kms_posCrustToMantle/I23K_MCMC.png'
print('\n    I23K_wtRFVphHV__4pt9kms_posCrustToMantle :')
f=Image.open(newf)
f.thumbnail(imgsize,Image.ANTIALIAS)
display(f)
print('\n\n\n\n')
```

```python
newff=pltdir+'/I23K_wtRFVphHV__4pt9kms_posCrustToMantle/I23K_IterVsMisfit.png'
f=Image.open(newff)
f.thumbnail(imgsize1,Image.ANTIALIAS)
display(f)
print('\n\n')


newf=pltdir+'/CAST_wtRFVphHV__4pt9kms_posCrustToMantle/CAST_MCMC.png'
sm=Image.open(newf)
sm.thumbnail(imgsize,Image.ANTIALIAS)
print('\n   CAST_wtRFVphHV__4pt9kms_posCrustToMantle :')
display(sm)

newff=pltdir+'/CAST_wtRFVphHV__4pt9kms_posCrustToMantle/CAST_IterVsMisfit.png'
sm=Image.open(newff)
sm.thumbnail(imgsize1,Image.ANTIALIAS)
display(sm)
print('\n\n')


####################################################
```

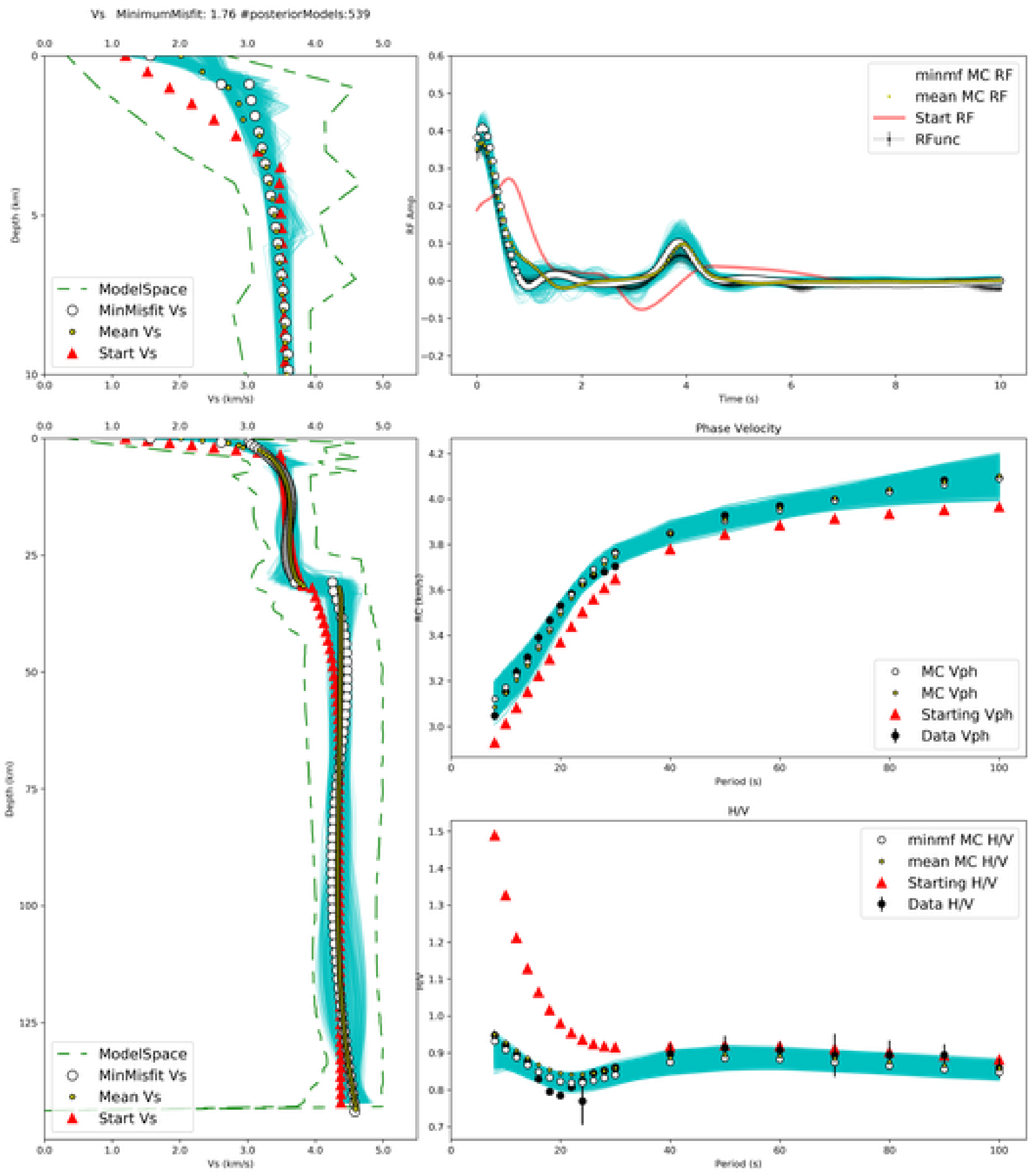
```
Joint inversion, weighted datasets

   I23K_wtRFVphHV__4pt9kms_posCrustToMantle :
```

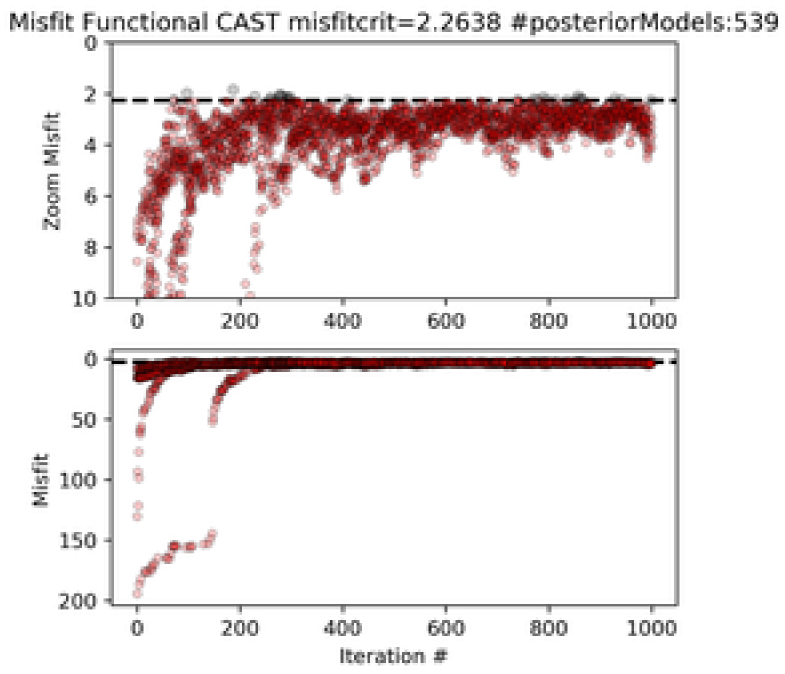


Vs MinimumMisfit: 1.56 #posteriorModels:251
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
MC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V



`CAST_wtRFVphHV__4pt9kms_posCrustToMantle :`

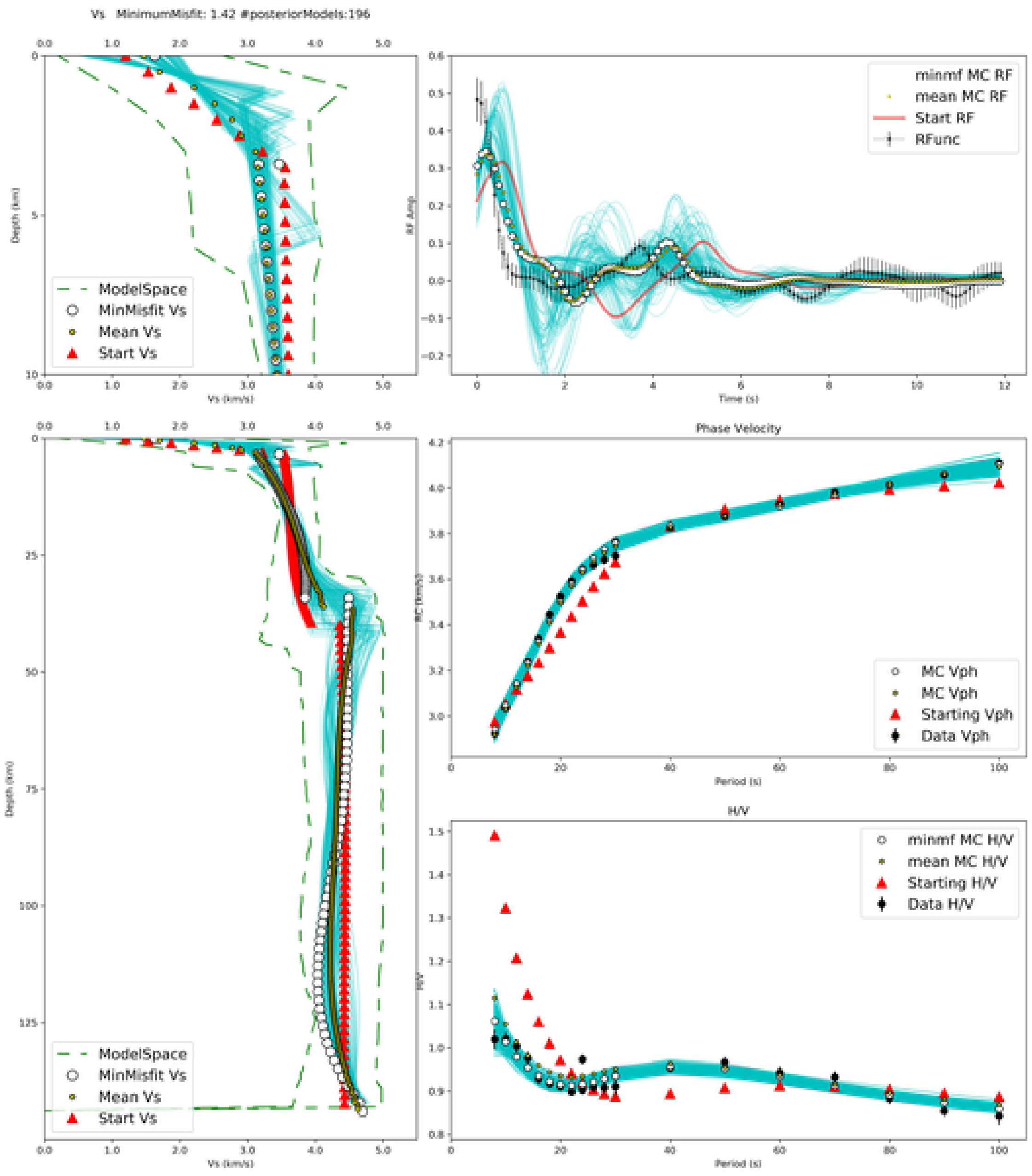

Vs MinimumMisfit: 1.76 #posteriorModels:539
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
MC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V

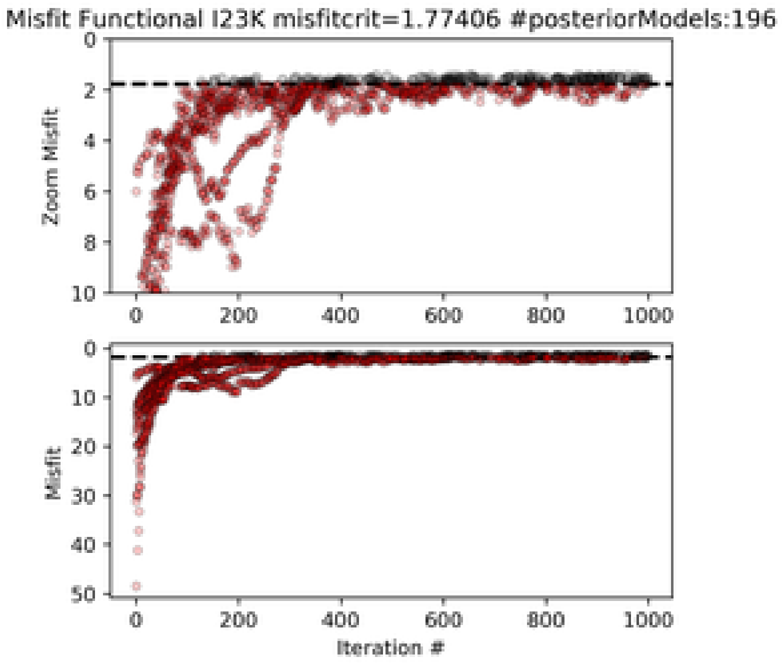


[59]:
```
pltdir='/uufs/chpc.utah.edu/common/home/u1015716/JointInversion_Tutorial/
↪MCMC_TestData'

imgsize=[600,600]
imgsize1=[300,300]


####################################################
# from Preset Code Option 2 #

print('Receiver function data NOT used in joint inversion')
print('weight H/V 2x over Vph')

newf=pltdir+'/I23K_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline/
↪I23K_MCMC.png'
sm=Image.open(newf)
sm.thumbnail(imgsize,Image.ANTIALIAS)
print('\n   I23K_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline :')
display(sm)
```

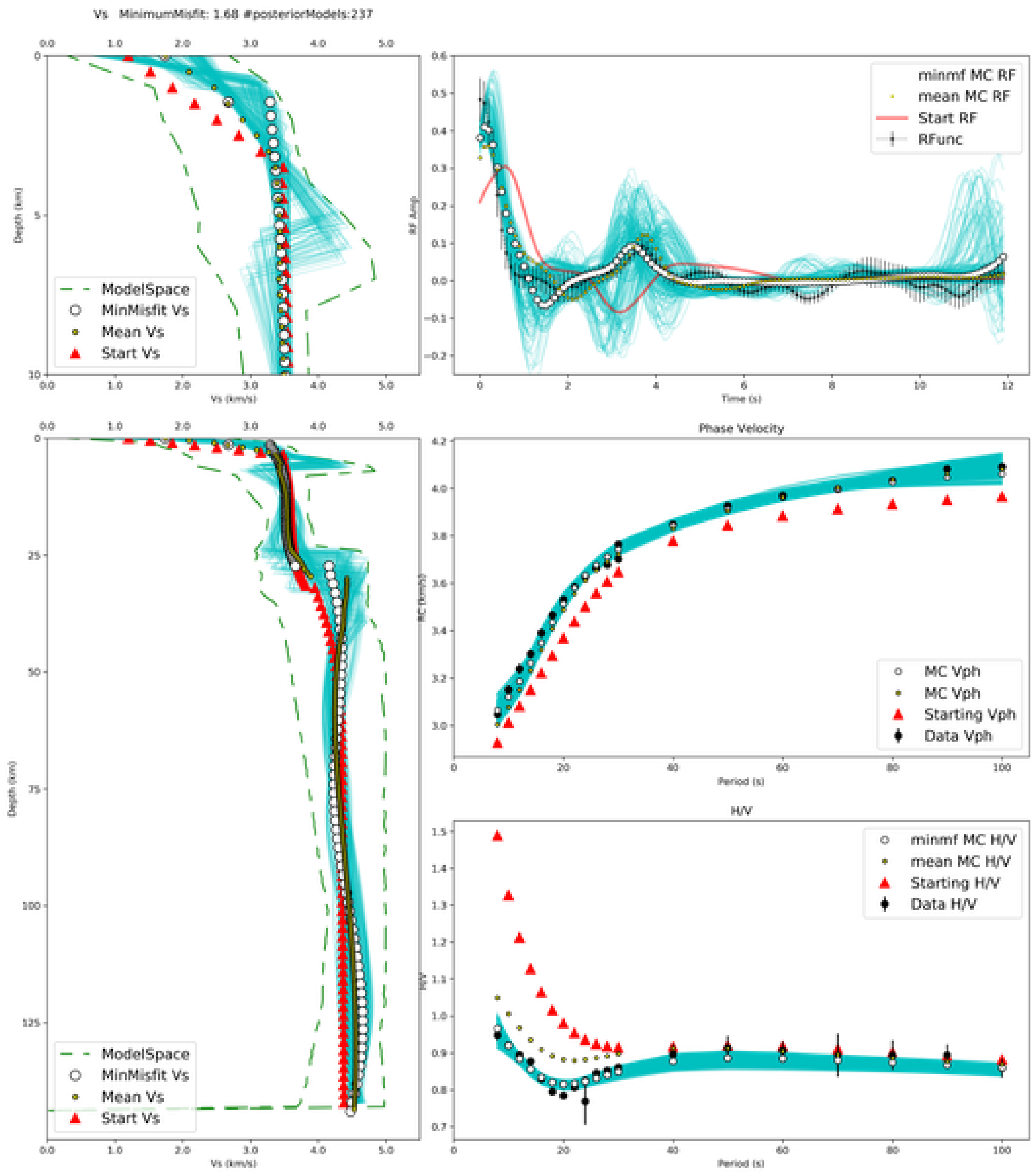
```
newff=pltdir+'/I23K_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline/
 ↪I23K_IterVsMisfit.png'
sm=Image.open(newff)
sm.thumbnail(imgsize1,Image.ANTIALIAS)
display(sm)
print('\n\n')


newf=pltdir+'/CAST_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline/
 ↪CAST_MCMC.png'
sm=Image.open(newf)
sm.thumbnail(imgsize,Image.ANTIALIAS)
print('\n   CAST_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline :')
display(sm)

newff=pltdir+'/CAST_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline/
 ↪CAST_IterVsMisfit.png'
sm=Image.open(newff)
sm.thumbnail(imgsize1,Image.ANTIALIAS)
display(sm)
print('\n\n')


####################################################
```

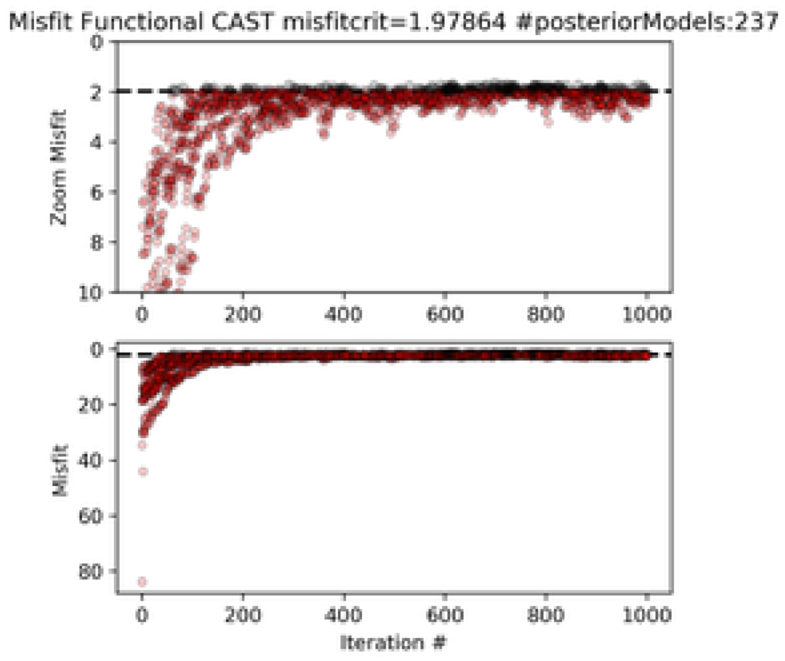
```
Receiver function data NOT used in joint inversion
weight H/V 2x over Vph

   I23K_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline :
```


Vs MinimumMisfit: 1.42 #posteriorModels:196
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
MC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V



`CAST_wtVphHV__4pt9kms_posCrustToMantle_1MsplineLT2Mspline :`

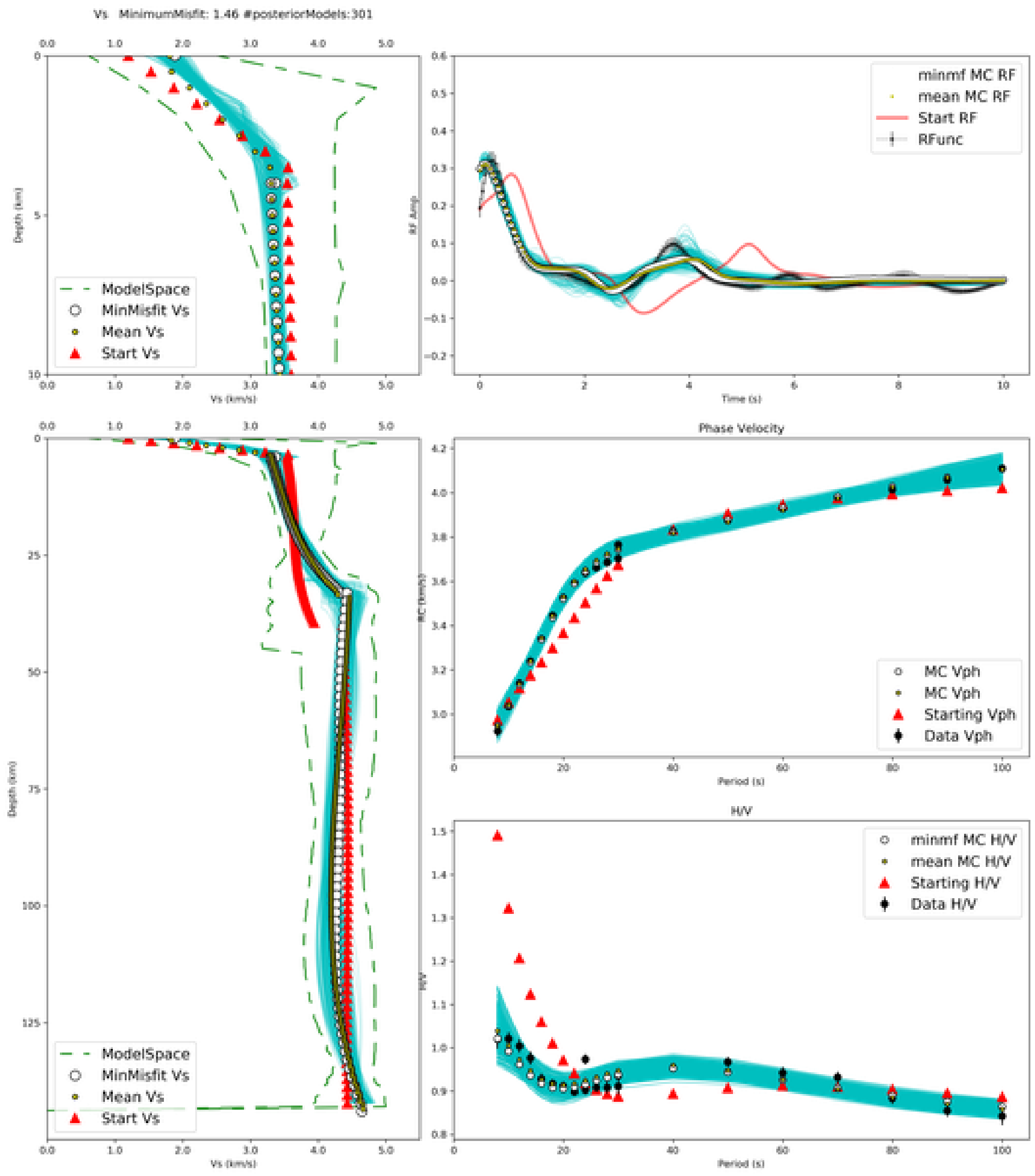

Vs MinimumMisfit: 1.68 #posteriorModels:237
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
RC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V

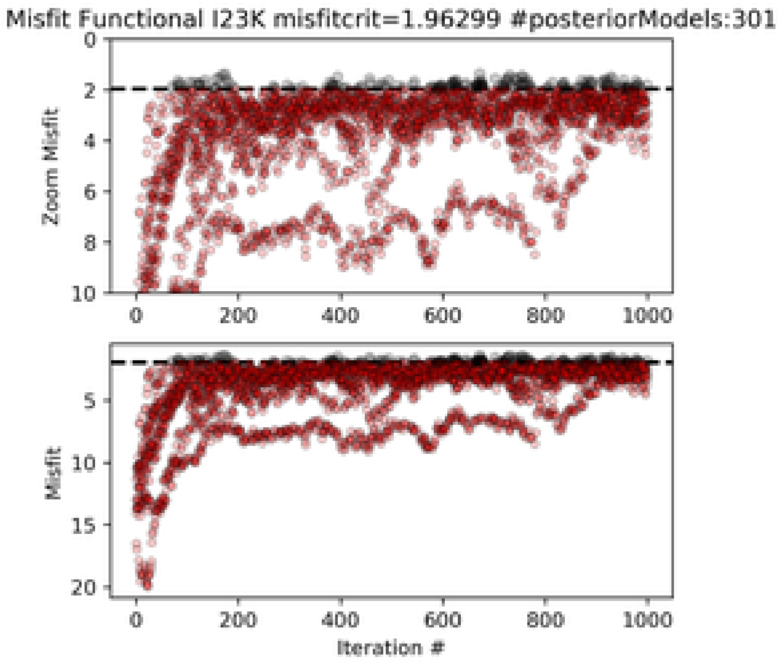


```
[60]: pltdir='/uufs/chpc.utah.edu/common/home/u1015716/JointInversion_Tutorial/
       ↪MCMC_TestData'

      imgsize=[600,600]
      imgsize1=[300,300]


      ####################################################
      # from Preset Code Option 3 #

      print('All data used with equal weighting in joint inversion')

      newf=pltdir+'/I23K_RFVphHV__4pt9kms_posCrustToMantle/I23K_MCMC.png'
      sm=Image.open(newf)
      sm.thumbnail(imgsize,Image.ANTIALIAS)
      print('\n    I23K_RFVphHV__4pt9kms_posCrustToMantle :')
      display(sm)

      newff=pltdir+'/I23K_RFVphHV__4pt9kms_posCrustToMantle/I23K_IterVsMisfit.png'
      sm=Image.open(newff)
      sm.thumbnail(imgsize1,Image.ANTIALIAS)
```

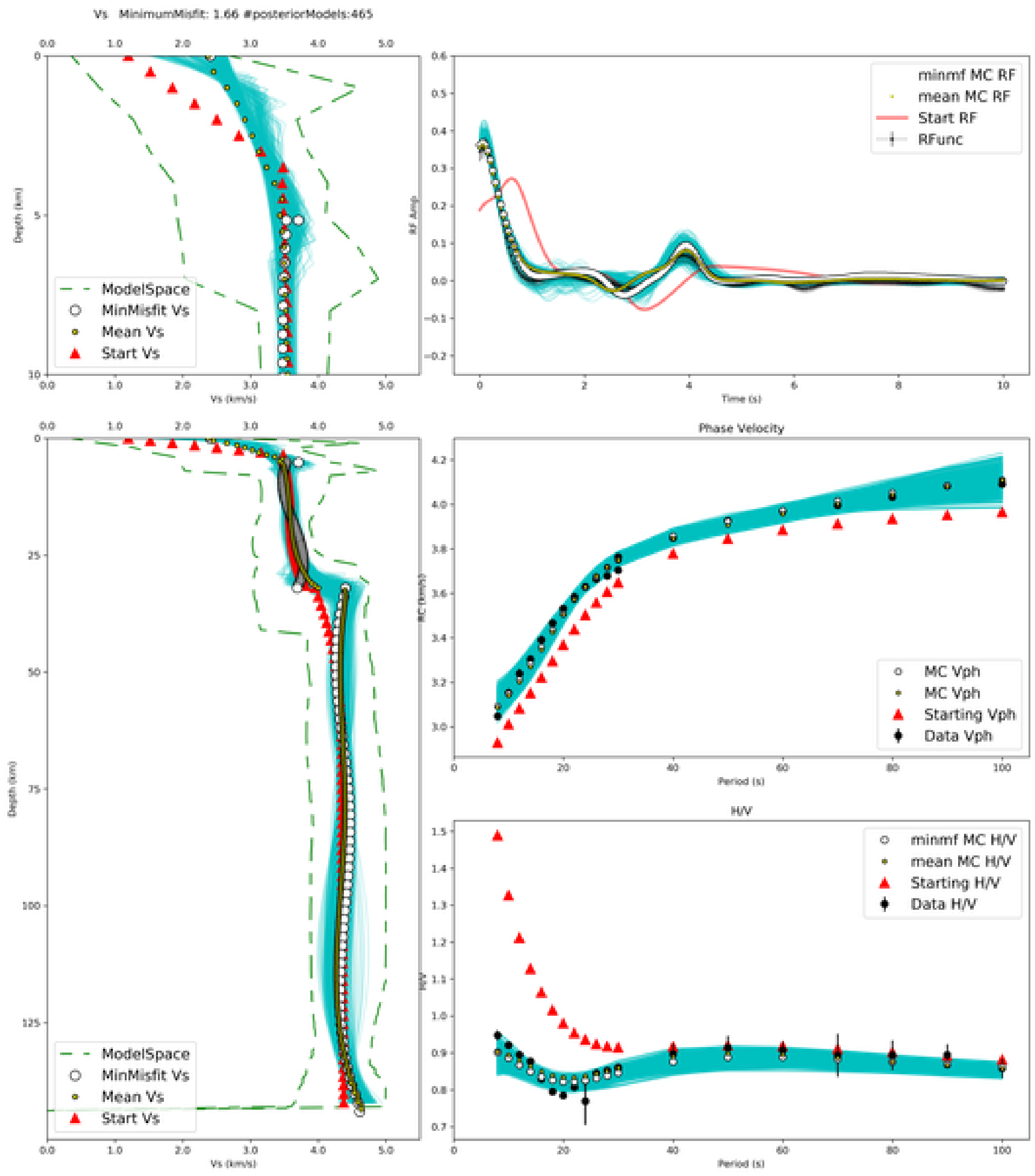
```python
display(sm)
print('\n\n')


newf=pltdir+'/CAST_RFVphHV__4pt9kms_posCrustToMantle/CAST_MCMC.png'
sm=Image.open(newf)
sm.thumbnail(imgsize,Image.ANTIALIAS)
print('\n   CAST_RFVphHV__4pt9kms_posCrustToMantle :')
display(sm)

newff=pltdir+'/CAST_RFVphHV__4pt9kms_posCrustToMantle/CAST_IterVsMisfit.png'
sm=Image.open(newff)
sm.thumbnail(imgsize1,Image.ANTIALIAS)
display(sm)
print('\n\n')


####################################################
```

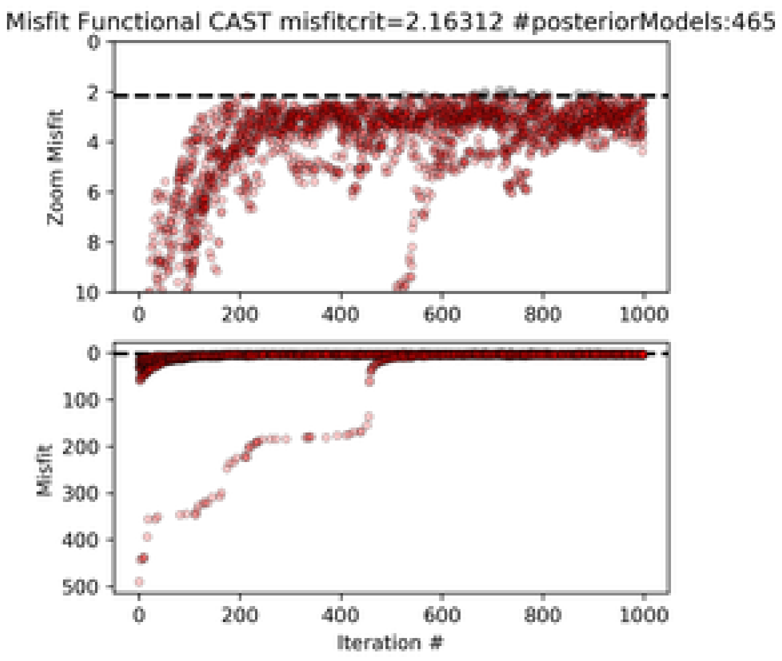
```
All data used with equal weighting in joint inversion

   I23K_RFVphHV__4pt9kms_posCrustToMantle :
```


Vs MinimumMisfit: 1.46 #posteriorModels:301
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
PC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V



`CAST_RFVphHV__4pt9kms_posCrustToMantle :`

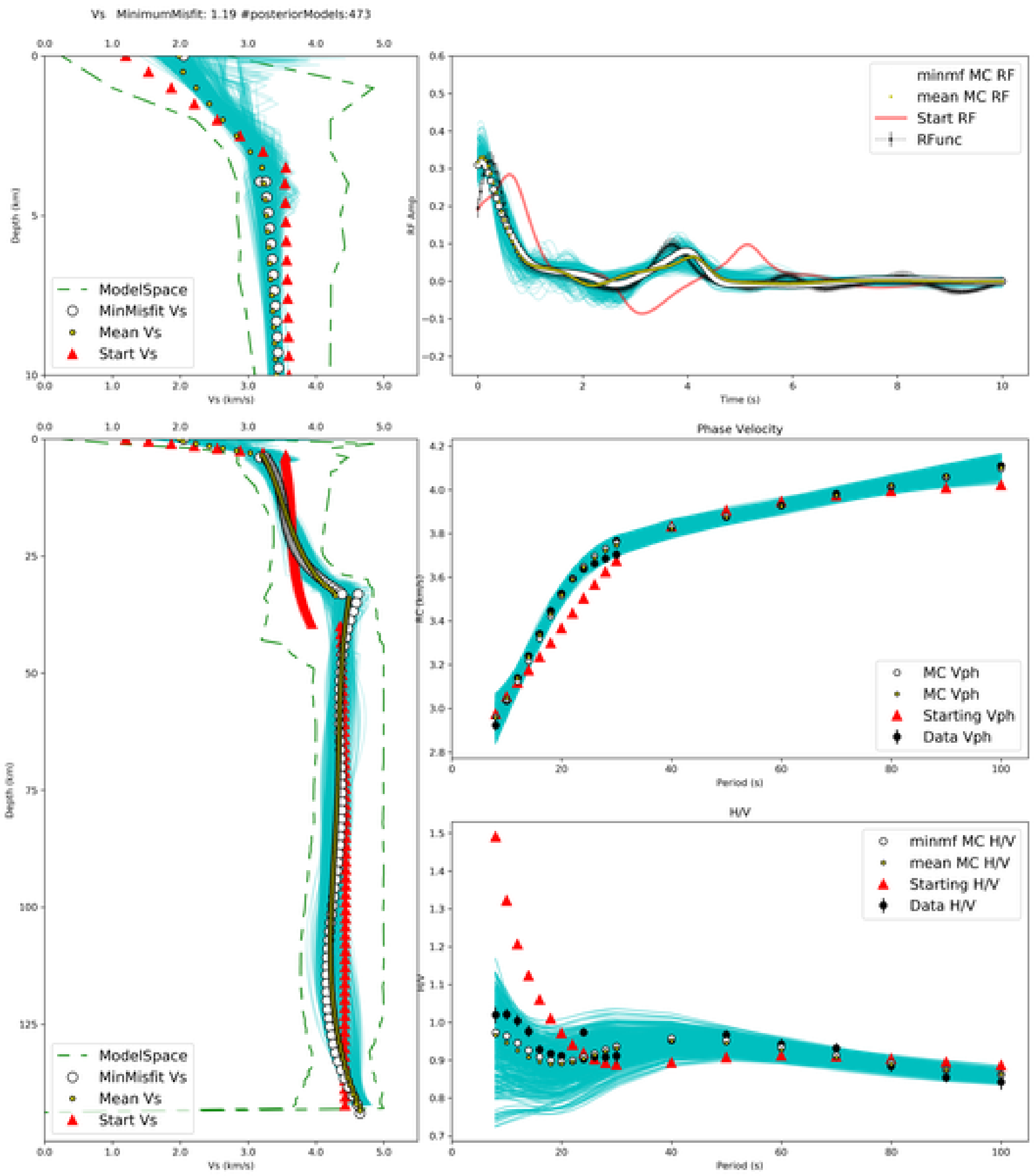

Vs MinimumMisfit: 1.66 #posteriorModels:465
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
MC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V



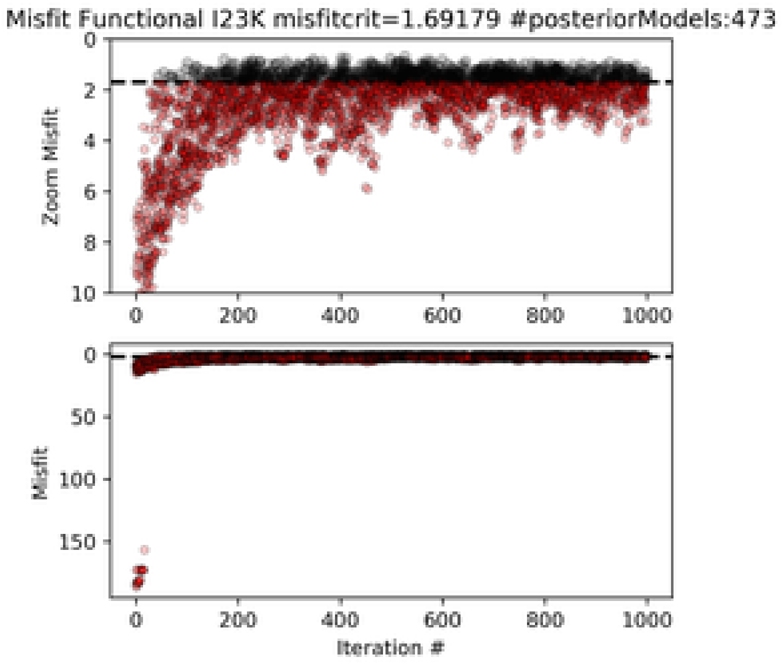
```
[61]: pltdir='/uufs/chpc.utah.edu/common/home/u1015716/JointInversion_Tutorial/
       ↪MCMC_TestData'

      imgsize=[600,600]
      imgsize1=[300,300]



      ####################################################
      # from Preset Code Option 4 #

      print('Only Vph & Receiver function data are used with equal weighting in joint␣
       ↪inversion')

      newf=pltdir+'/I23K_RFVph__4pt9kms_posCrustToMantle_SedIncreasing/I23K_MCMC.png'
      sm=Image.open(newf)
      sm.thumbnail(imgsize,Image.ANTIALIAS)
      print('\n    I23K_RFVph__4pt9kms_posCrustToMantle_SedIncreasing :')
      display(sm)

      newff=pltdir+'/I23K_RFVph__4pt9kms_posCrustToMantle_SedIncreasing/
       ↪I23K_IterVsMisfit.png'
```

```
sm=Image.open(newff)
sm.thumbnail(imgsize1,Image.ANTIALIAS)
display(sm)
print('\n\n')


newf=pltdir+'/CAST_RFVph__4pt9kms_posCrustToMantle_SedIncreasing/CAST_MCMC.png'
sm=Image.open(newf)
sm.thumbnail(imgsize,Image.ANTIALIAS)
print('\n   CAST_RFVph__4pt9kms_posCrustToMantle_SedIncreasing :')
display(sm)

newff=pltdir+'/CAST_RFVph__4pt9kms_posCrustToMantle_SedIncreasing/
 ↪CAST_IterVsMisfit.png'
sm=Image.open(newff)
sm.thumbnail(imgsize1,Image.ANTIALIAS)
display(sm)
print('\n\n')


####################################################
```

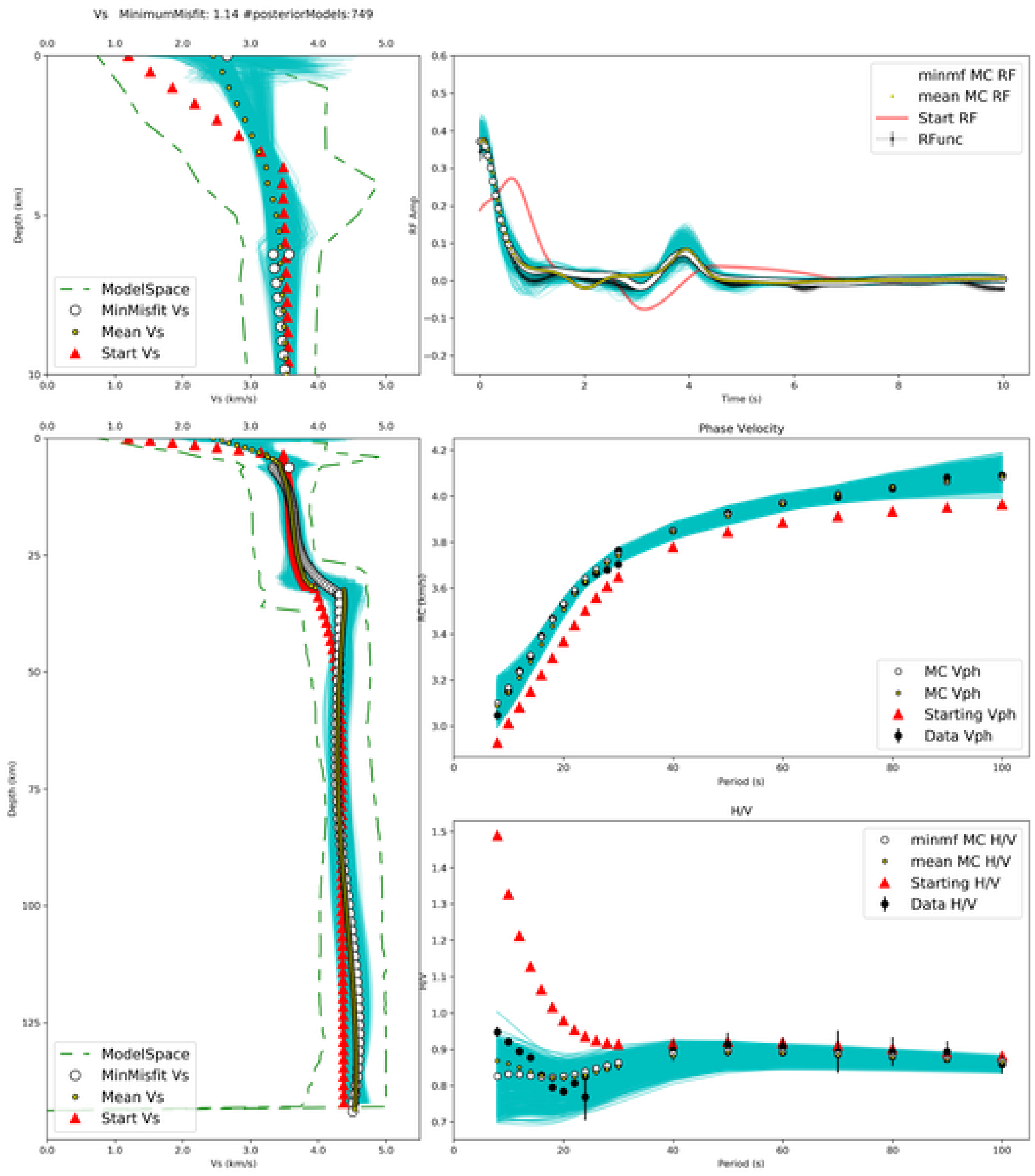
```
Only Vph & Receiver function data are used with equal weighting in joint
inversion

   I23K_RFVph__4pt9kms_posCrustToMantle_SedIncreasing :
```



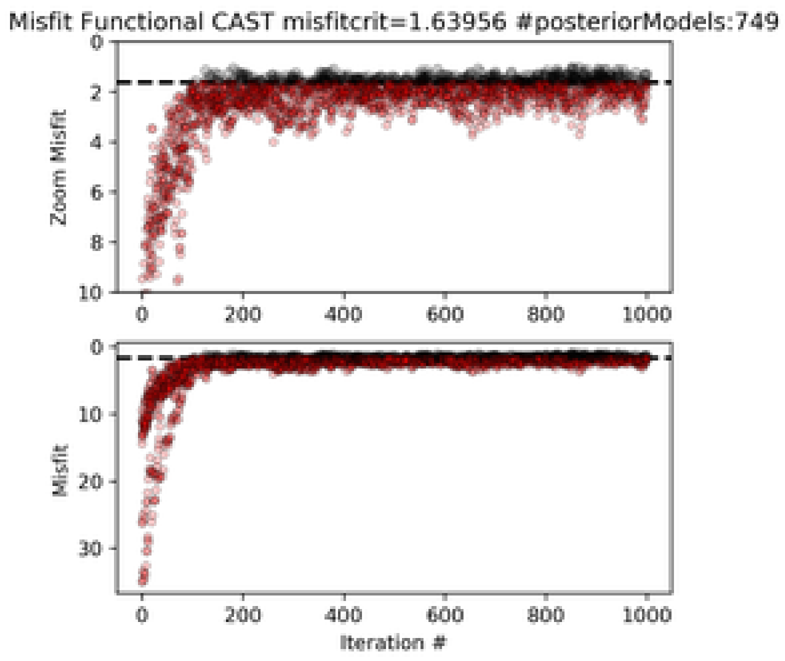
Vs MinimumMisfit: 1.19 #posteriorModels:473
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
RC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V



`CAST_RFVph__4pt9kms_posCrustToMantle_SedIncreasing :`



Vs MinimumMisfit: 1.14 #posteriorModels:749
Depth (km)
Vs (km/s)
ModelSpace
MinMisfit Vs
Mean Vs
Start Vs
RF Amp
Time (s)
minmf MC RF
mean MC RF
Start RF
RFunc
Phase Velocity
MC (km/s)
Period (s)
MC Vph
MC Vph
Starting Vph
Data Vph
H/V
minmf MC H/V
mean MC H/V
Starting H/V
Data H/V


Misfit Functional CAST misfitcrit=1.63956 #posteriorModels:749
Zoom Misfit
Misfit
Iteration #

```
[ ]:
```