Introductory session
Alignment Projects

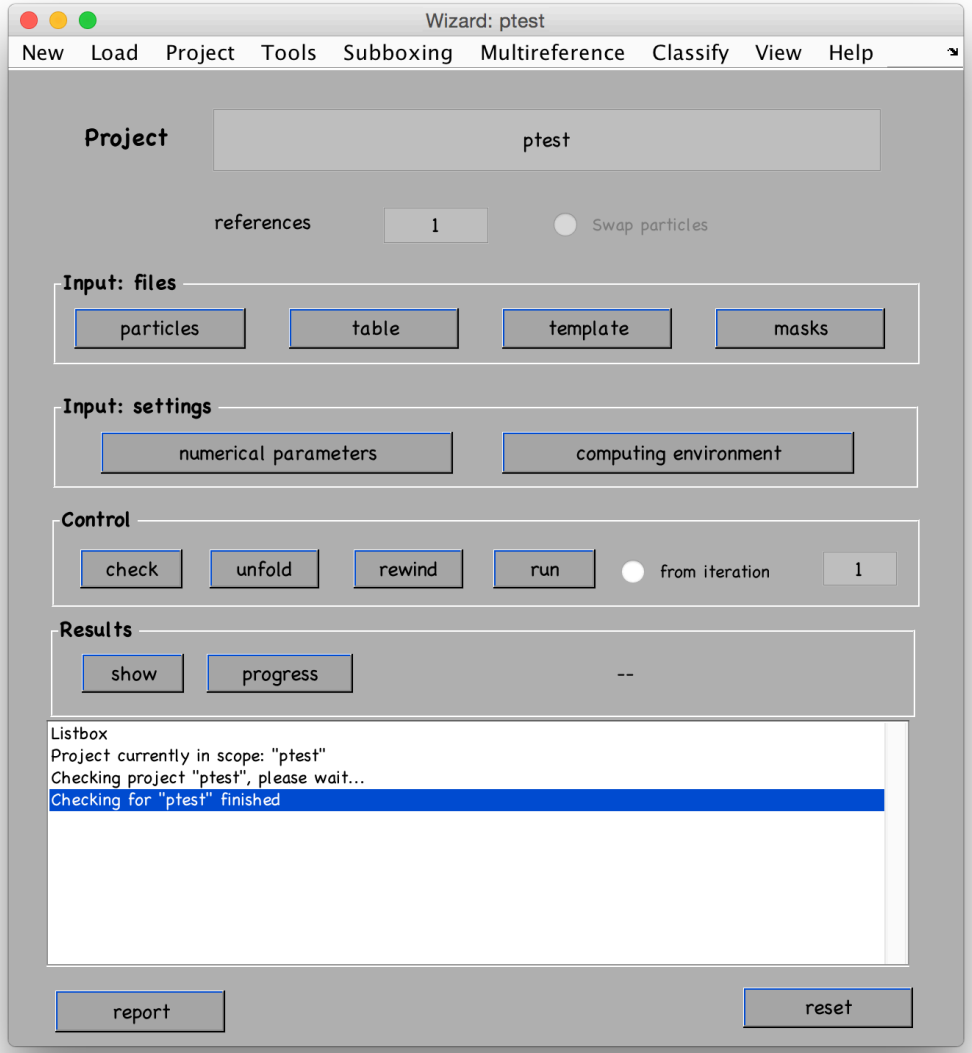
Let us create a tutorial that already contains an alignment project ready for execution:

```
>> dtutorial mytest –p ptest
```

It will take a couple of minutes. The resulting project `ptest` can be opened with:

```
>> dcp ptest
```

This Wizard GUI will guide you through the different steps in the process of project creation.



The *Dynamo* Wizard GUI

Basic project creation

The Wizard will open step by step different GUIs to complete the flow depicted on the right.

We will describe in the next slides the operations and considerations that the users needs to do on each step.

Main task to perform at each step

Additional tools

1. Link all the data and files
   a. Data particles
   b. Table(s)
   c. Initial template(s)
   d. Mask
      I. Alignment
      II. Classification
      III. Fourier content of templates
      IV. FSC Computation

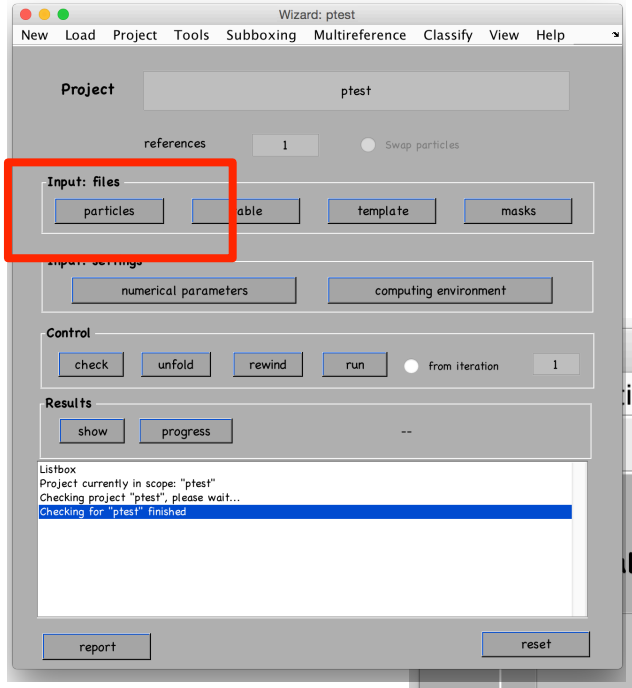
2. Numerical settings
   a. Angular scan
   b. Limit shifts
   c. Band pass parameters
   d. Cross correlation threshold

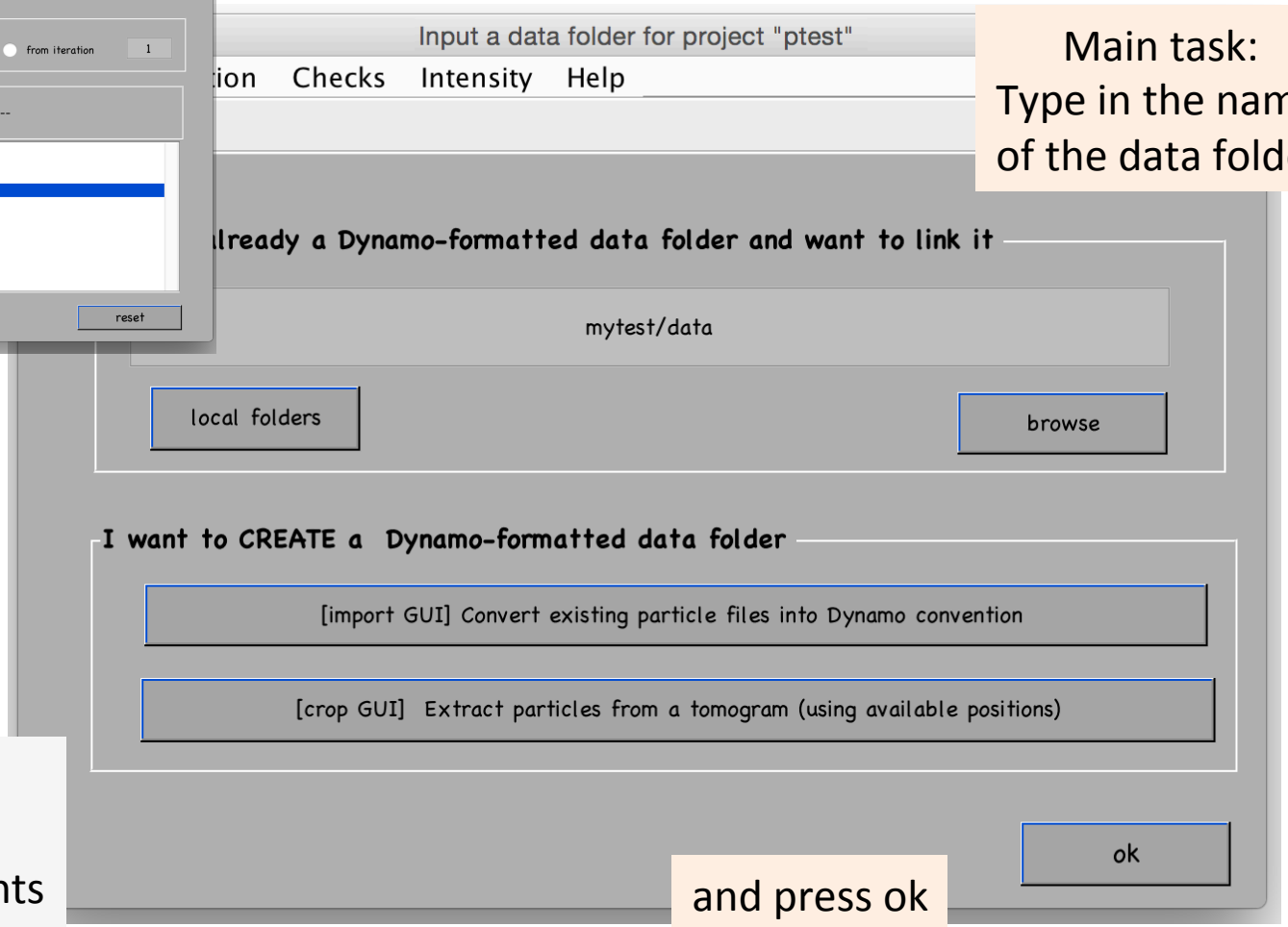
3. Indicate a hardware system:
   number of cores,

4. Check and run

5. Scan execution status and visualize results

Data particles



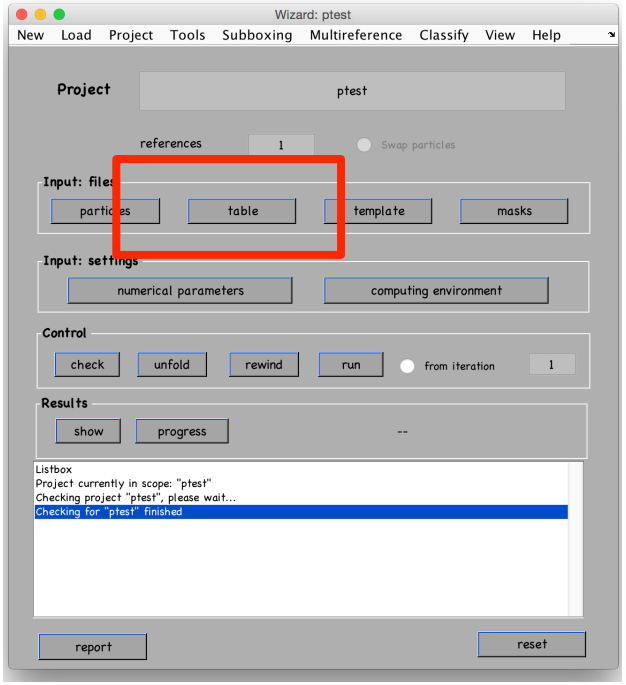


Main task:
Type in the name of the data folder

and press ok

Additional tools:

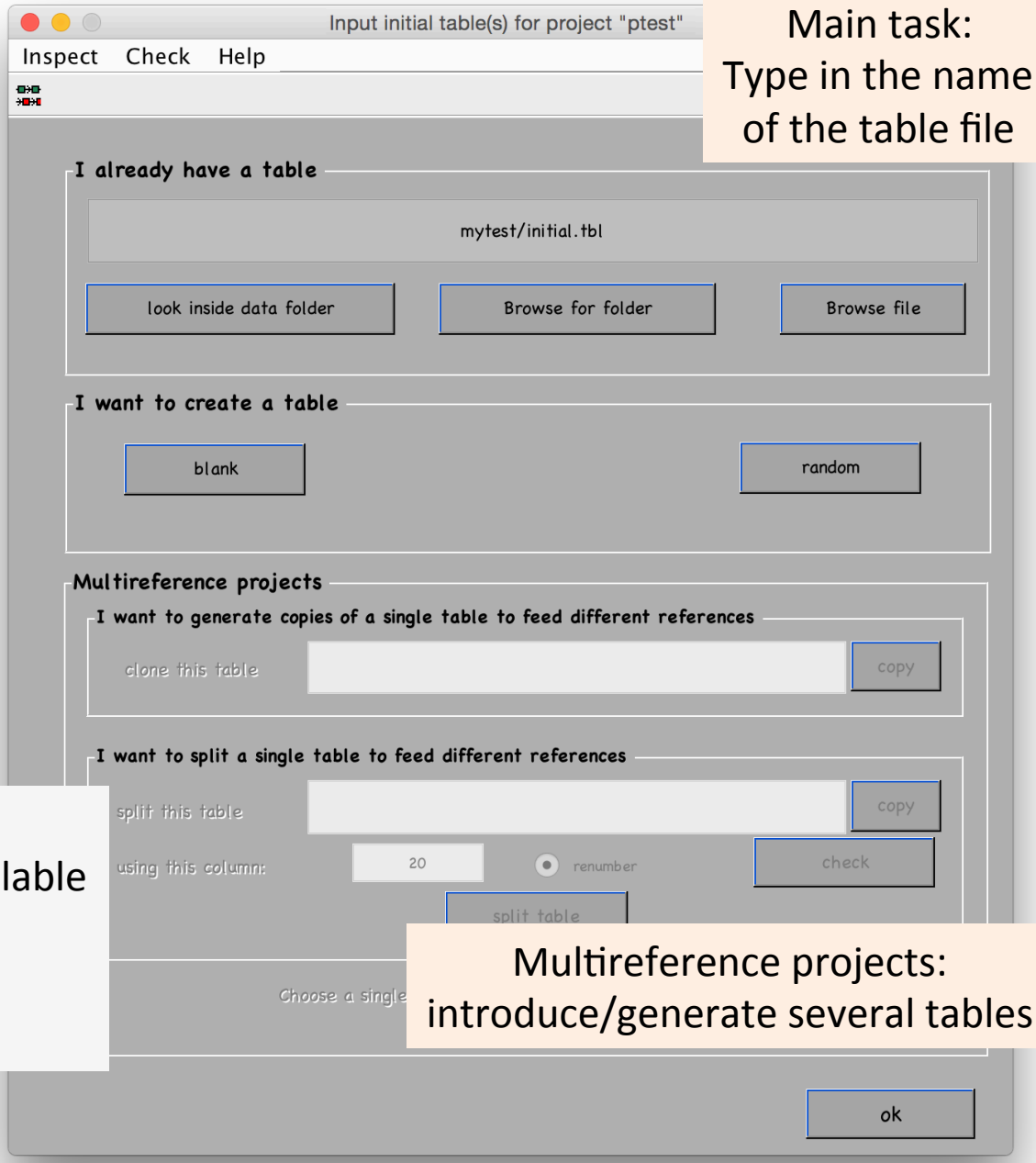
- Check contents
- Visualize contents
- Import formats



Table

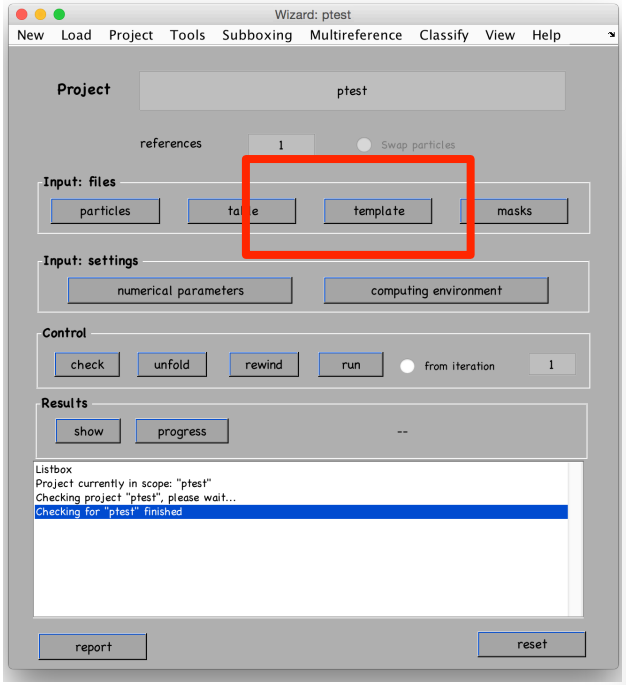
Additional tools:

- Create blank table if none available
- Visualize contents
- Import formats
- Check coherence with data



Main task:
Type in the name of the table file

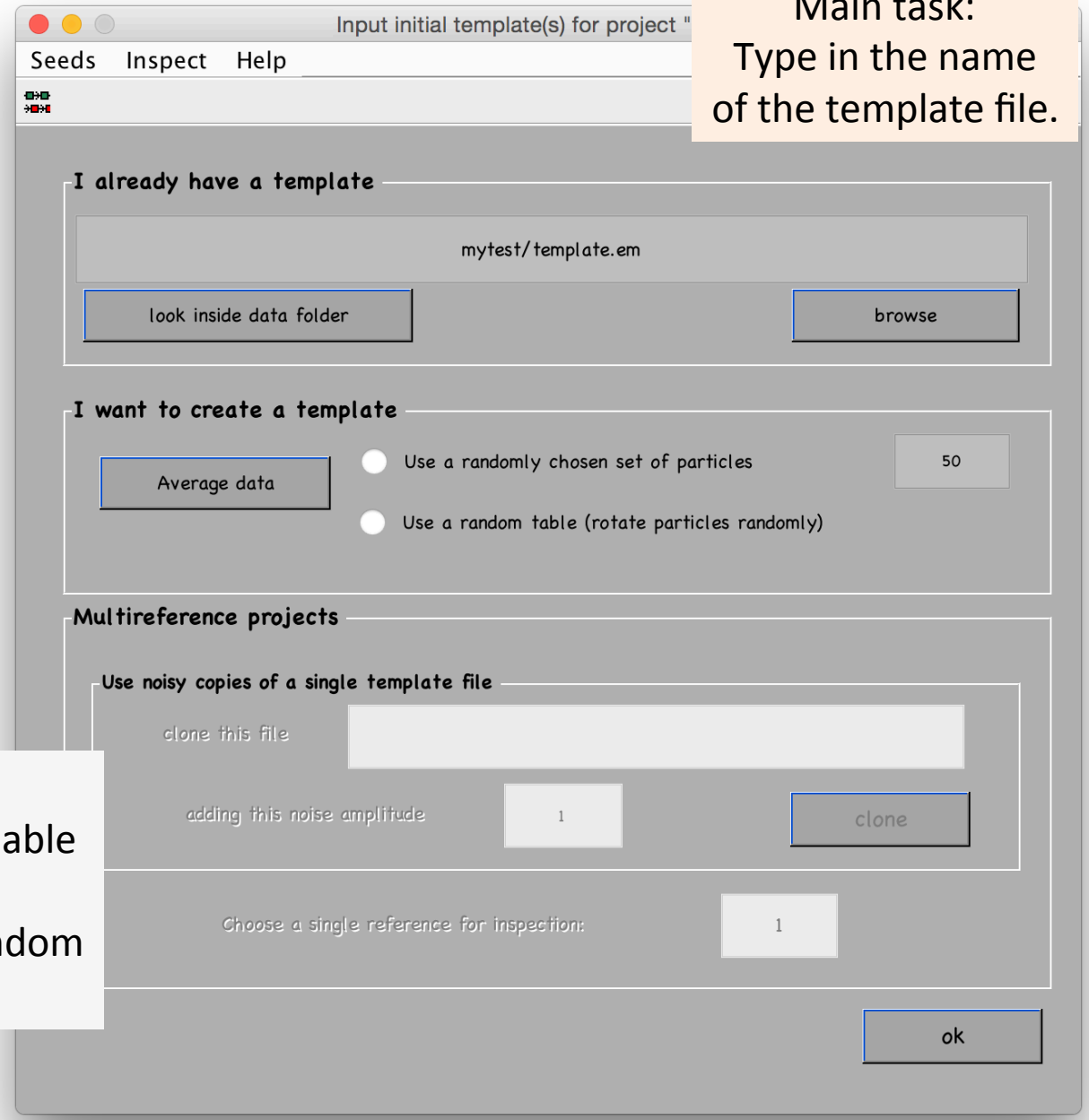
Multireference projects:
introduce/generate several tables



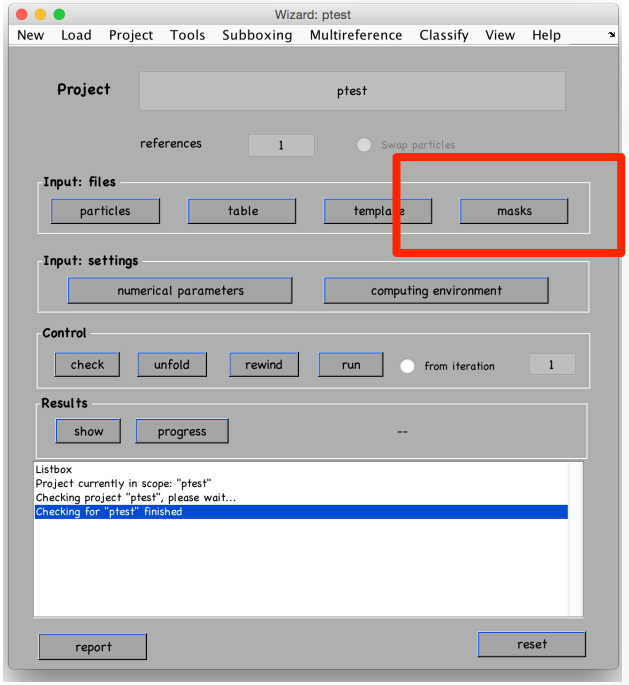
Template

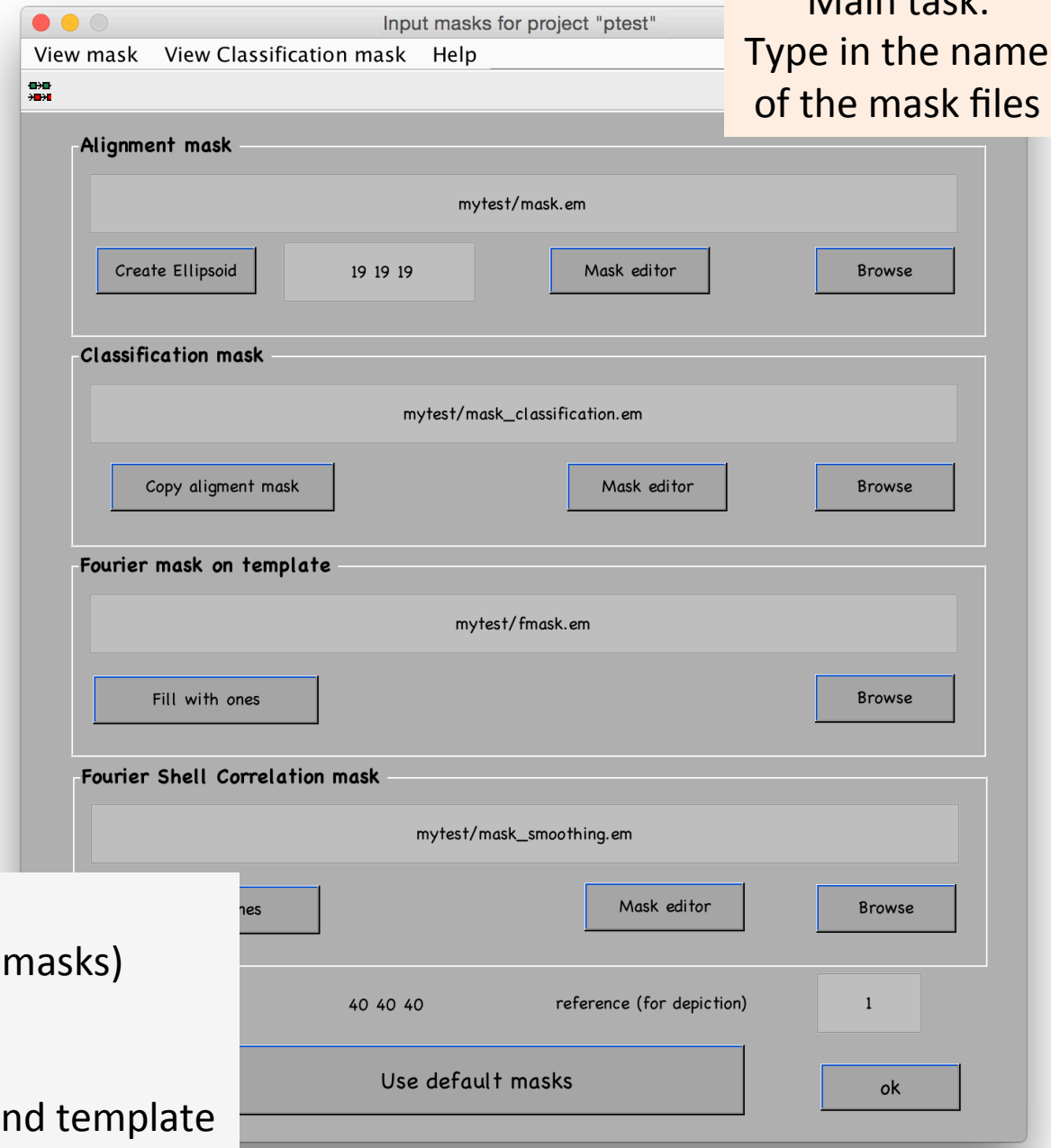
Additional tools:

- Create template if none available
- Visualize contents
- Multireference: generate random copies to be used as seeds



Main task:
Type in the name of the template file.





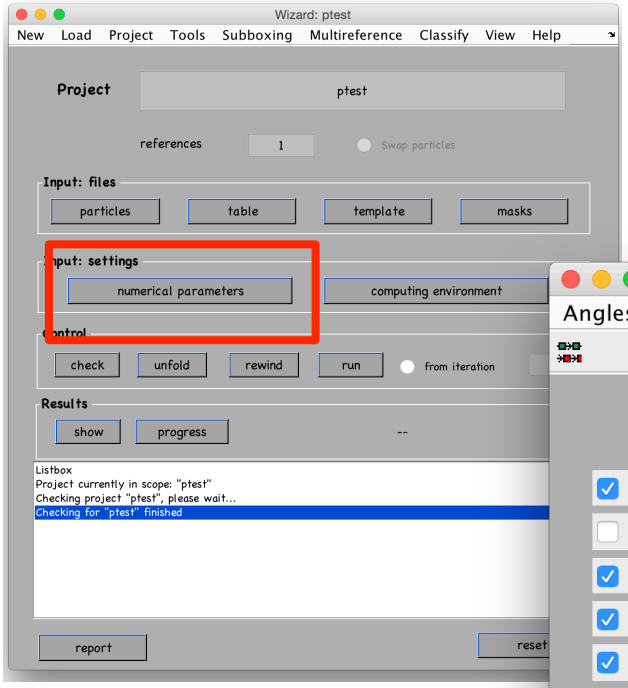
Main task:
Type in the name of the mask files

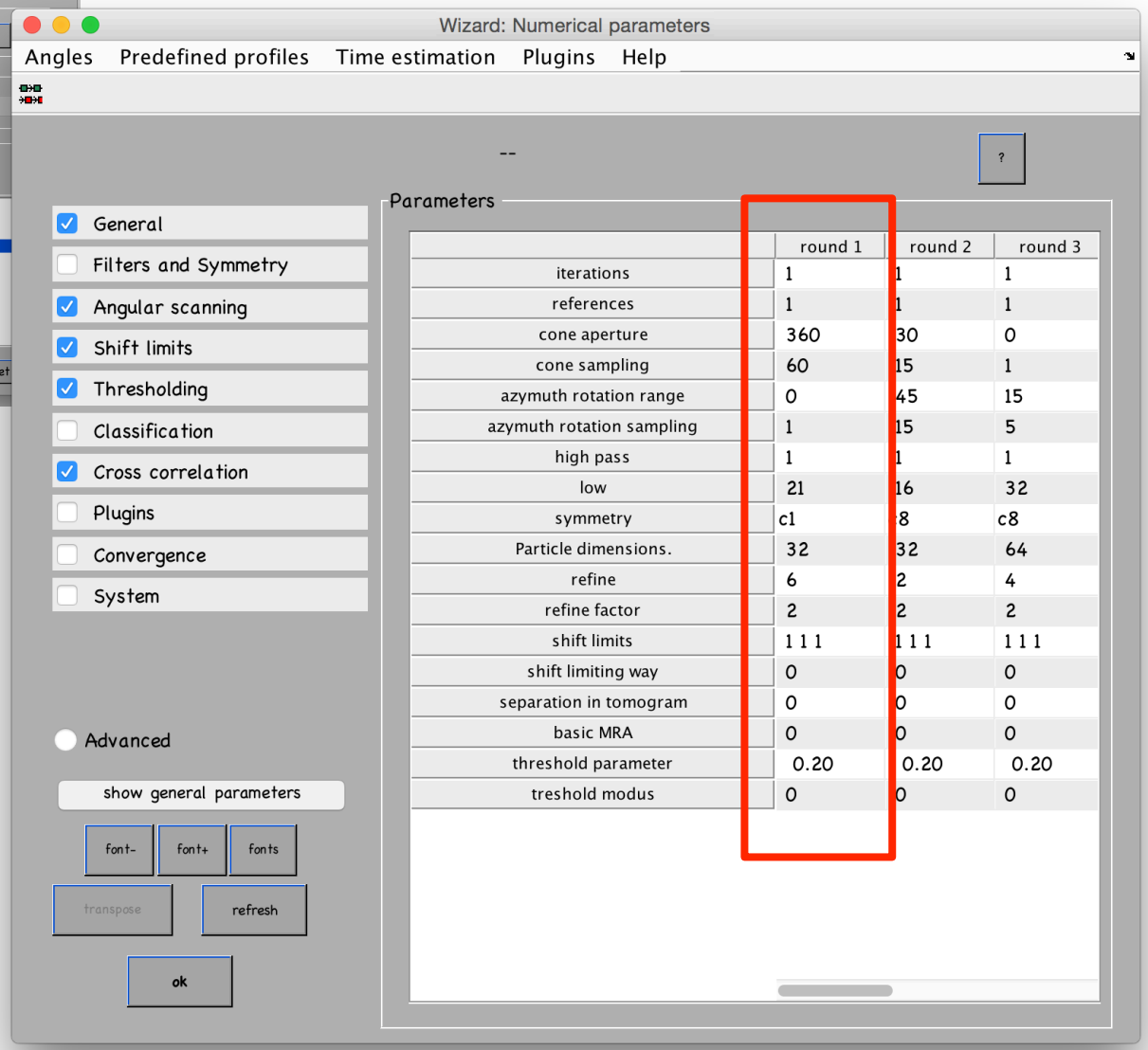
Masks

Additional tools:

- Generate defaults (spherical masks)
- Visualize contents
- Generate customized masks
- Check coherence with data and template

Numerical settings





	round 1	round 2	round 3
iterations	1	1	1
references	1	1	1
cone aperture	360	30	0
cone sampling	60	15	1
azymuth rotation range	0	45	15
azymuth rotation sampling	1	15	5
high pass	1	1	1
low	21	16	32
symmetry	c1	c8	c8
Particle dimensions.	32	32	64
refine	6	2	4
refine factor	2	2	2
shift limits	1 1 1	1 1 1	1 1 1
shift limiting way	0	0	0
separation in tomogram	0	0	0
basic MRA	0	0	0
threshold parameter	0.20	0.20	0.20
treshold modus	0	0	0

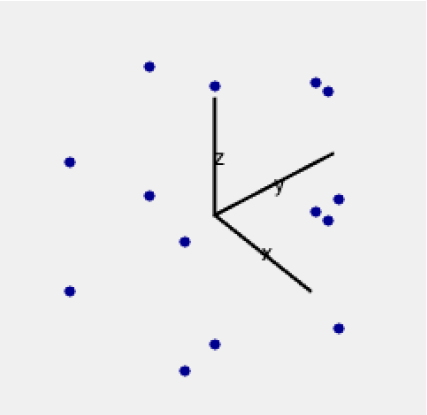
A round:
a set of iterations
ran with the same
parameters

The set of scanned angles at each round is defined by a set of six parameters

cone aperture	360	30	0
cone sampling	60	15	1
azymuth rotation range	0	45	15
azymuth rotation sampling	1	15	5
refine	6	2	4
refine factor	2	2	2

Cone aperture and cone sampling determine a set of axial orientations

cone aperture	360	30	0
cone sampling	60	15	1
azymuth rotation range	0	45	15
azymuth rotation sampling	1	15	5
refine	6	2	4
refine factor	2	2	2



"Generate orientations all around the sphere in angular intervals of 60 degrees"

Azymuth range and sampling determine a set of rotations around the axis

cone aperture	360	30	0
cone sampling	60	15	1
azymuth rotation range	0	45	15
azymuth rotation sampling	1	15	5
refine	6	2	4
refine factor	2	2	2

"No rotations around the axis"

This round is designed to perform a global search of directions without any azymuthal rotation.
It is intended to find one of the axes of the particles, without attempting to find how the particle is rotated around that axis, leaving that for later iterations.

Angular search is performed on a multilevel fashion.

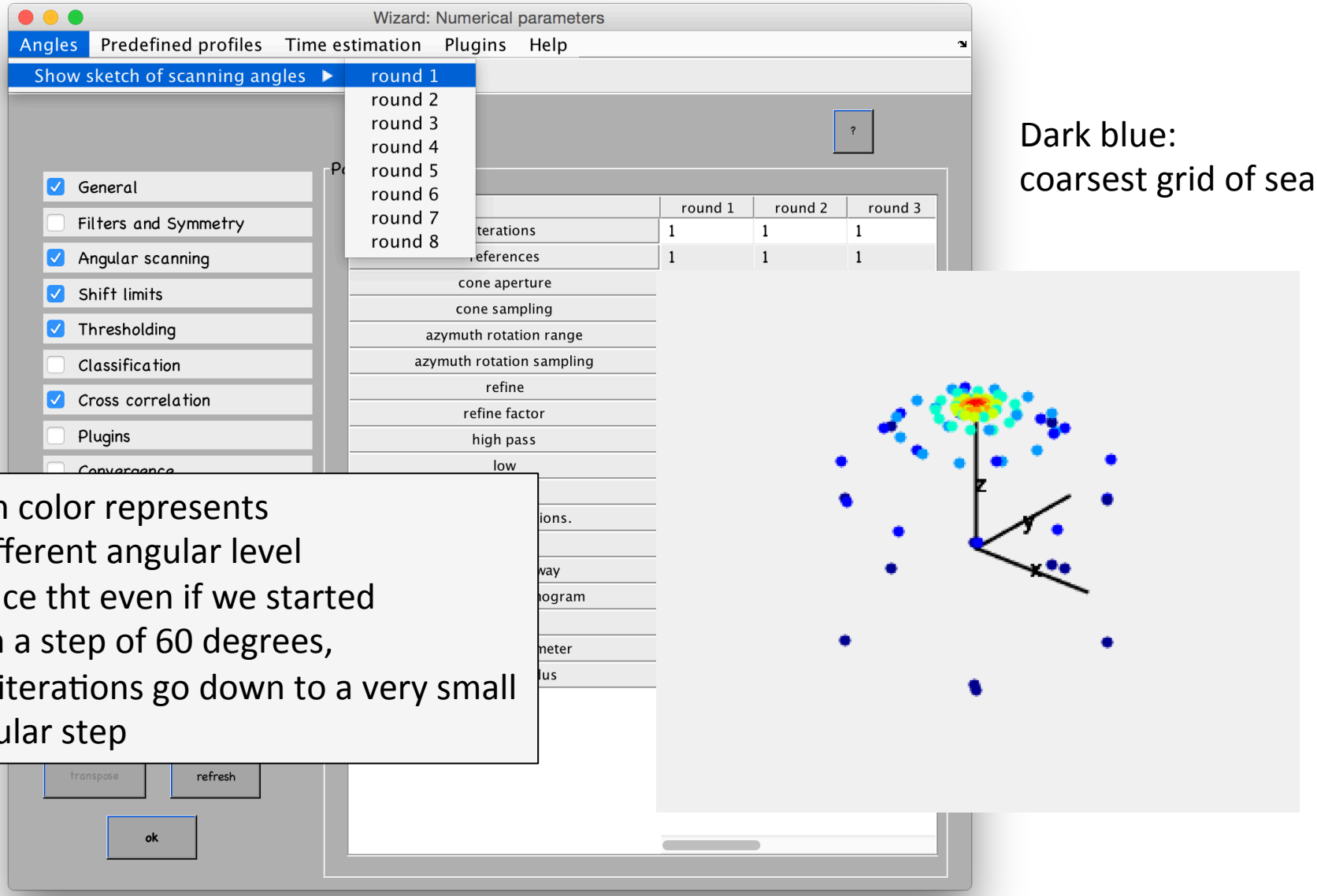
The previous parameters define the coarsest level of that hierarchy of searches

Now, refinement parameters create a local search around the best angular orientations found in the previous search level.

cone aperture	360	30	0
cone sampling	60	15	1
azymuth rotation range	0	45	15
azymuth rotation sampling	1	15	5
refine	6	2	4
refine factor	2	2	2

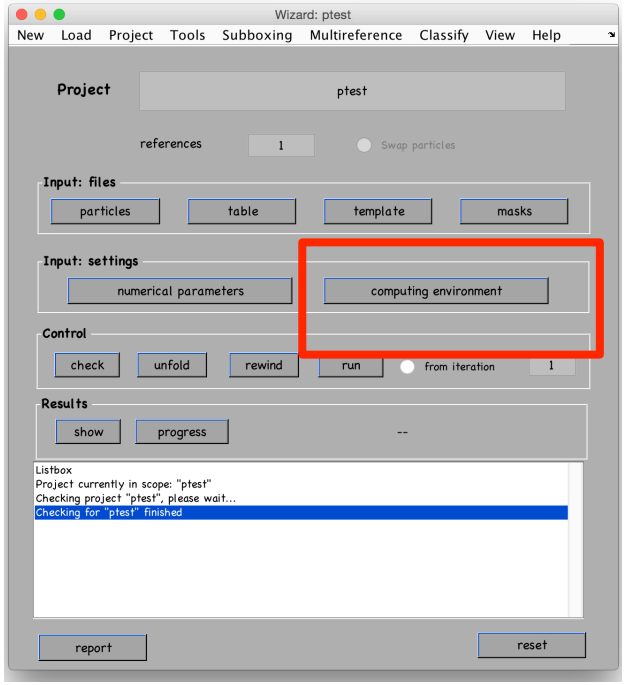
“refine six times after finding a maximum on a level.
The local scan range on each level is twice the scan interval of the previous level ”

You can show an sketch of the scan that you are designing (next slide)

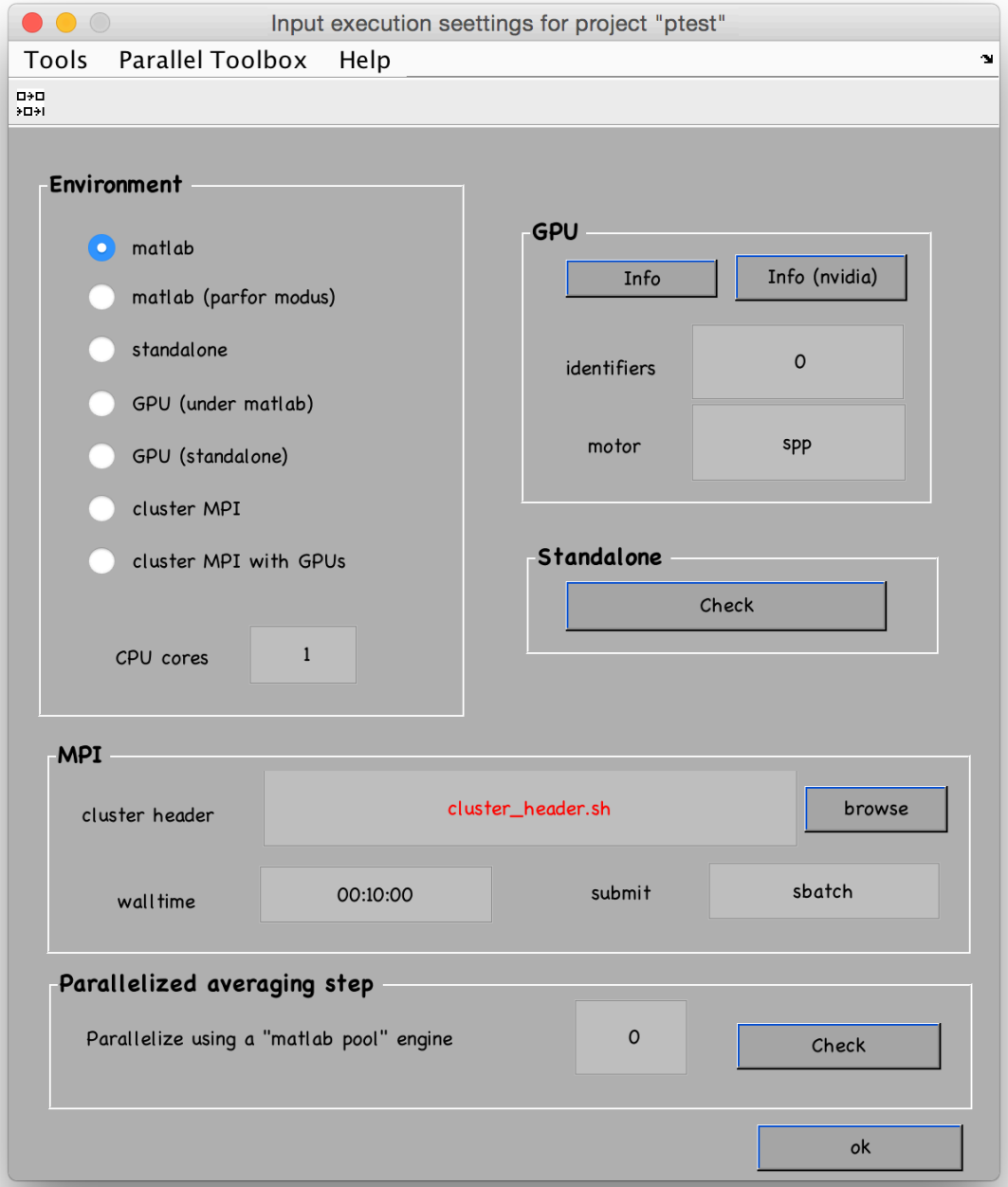


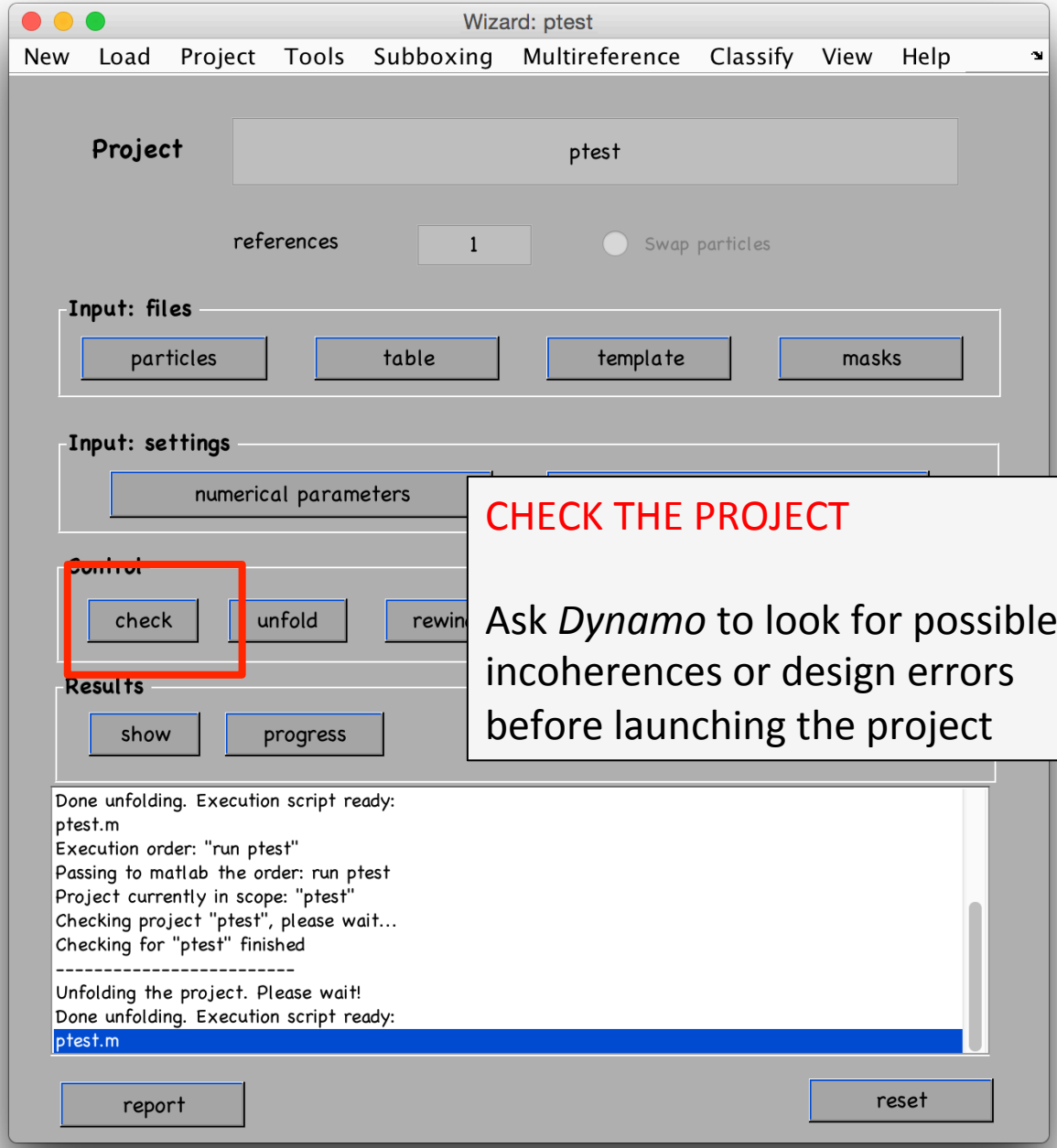
Dark blue:
coarsest grid of search

Each color represents
a different angular level
Notice tht even if we started
with a step of 60 degrees,
the iterations go down to a very small
angular step



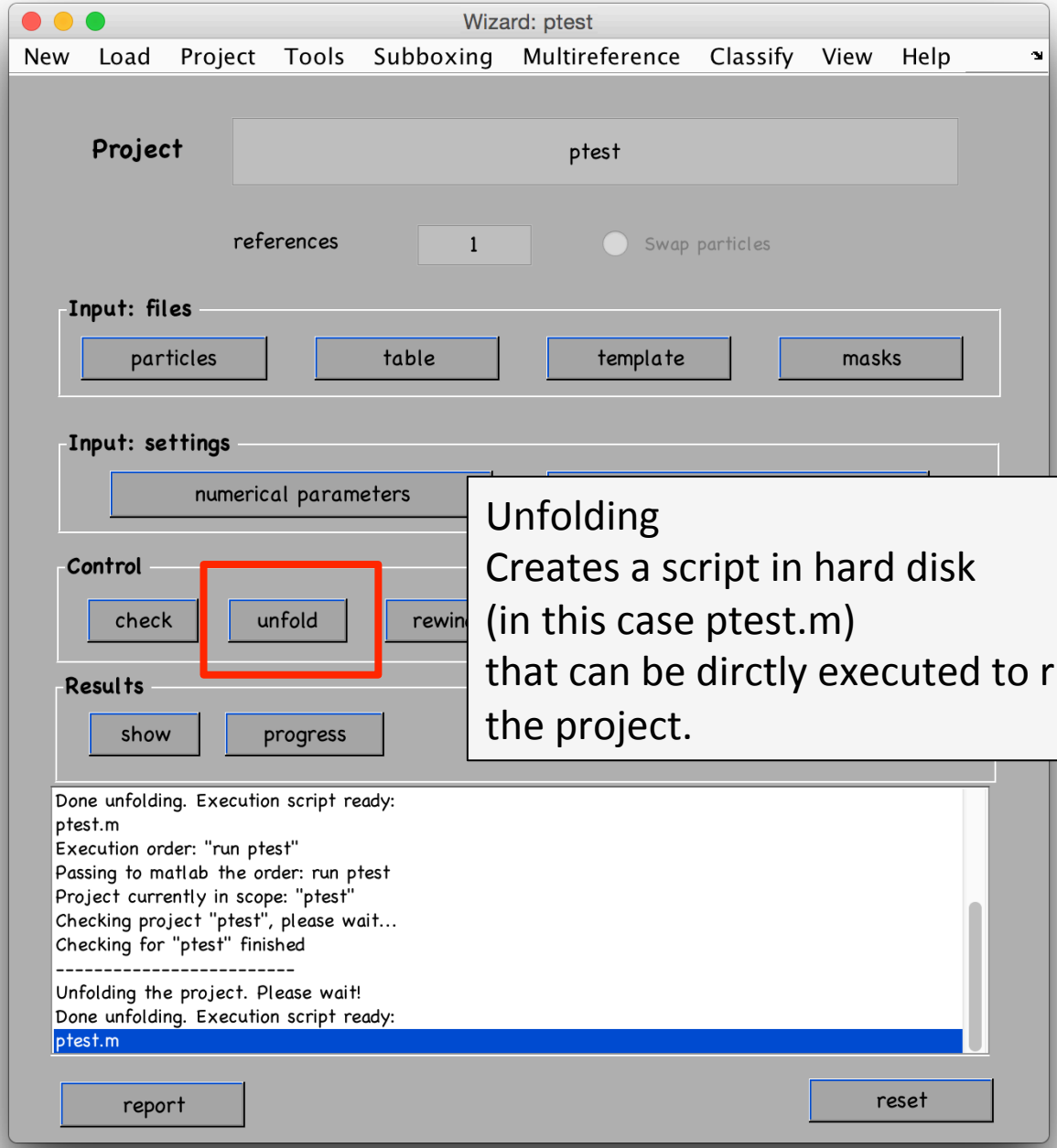
Select the hardware and environment where you want to run





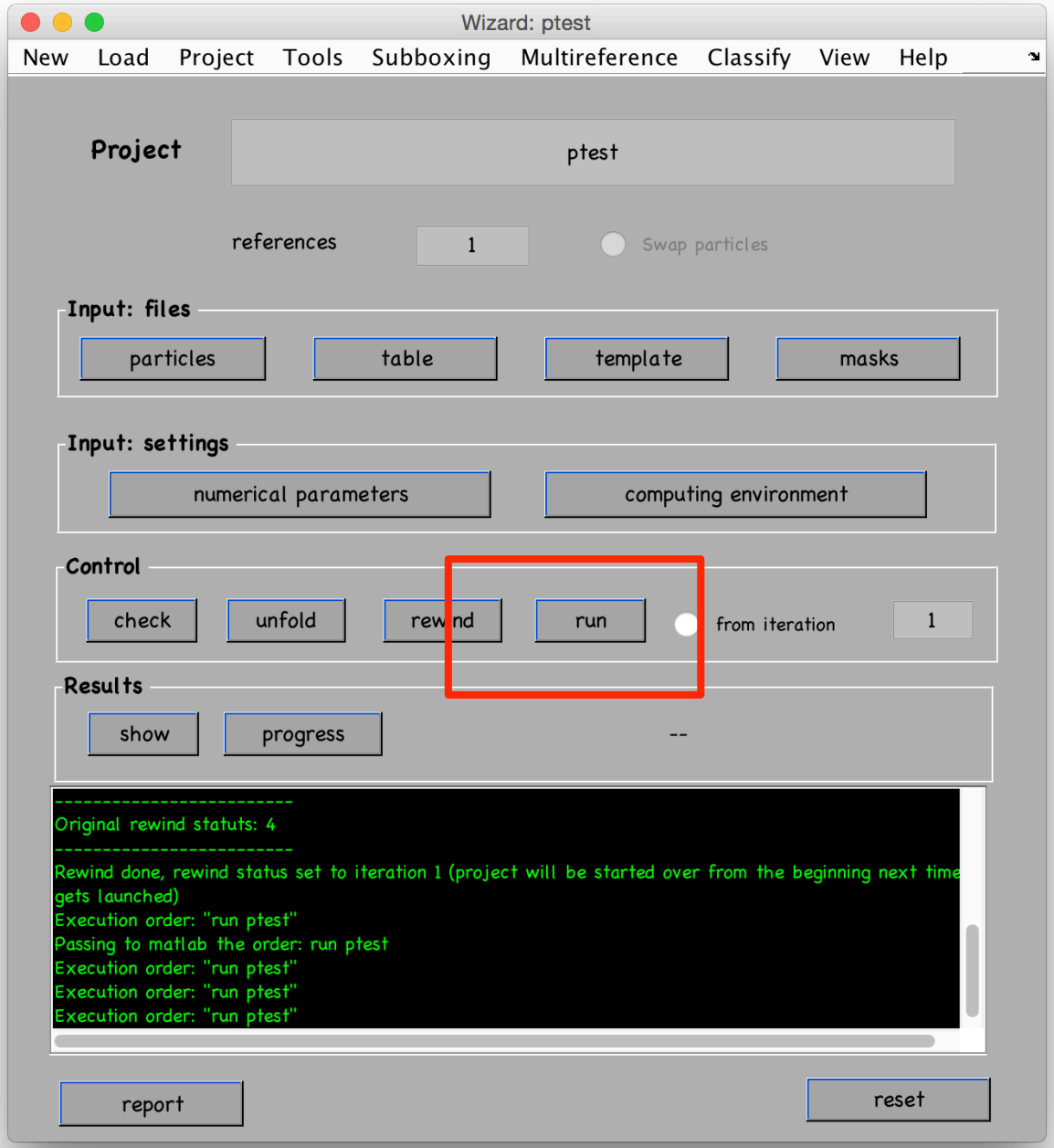
CHECK THE PROJECT

Ask *Dynamo* to look for possible incoherences or design errors before launching the project



Unfolding
Creates a script in hard disk
(in this case ptest.m)
that can be dirctly executed to run
the project.

If you press [run] the GUI will block while computing…



…. but the Matlab/Linux shell should report what is happening

MATLAB R2015a - academic use

HOME | PLOTS | APPS | SHORTCUTS | Extended | parser | dbquit | Search Documentation

New Script | New | Open | Find Files | Compare | Import Data | Save Workspace | New Variable | Open Variable | Clear Workspace | Analyze Code | Run and Time | Clear Commands | Simulink Library | Layout | Preferences | Set Path | Parallel | Help | Community | Request Support | Add-Ons

FILE | VARIABLE | CODE | SIMULINK | ENVIRONMENT | RESOURCES

/ ▸ Users ▸ casdanie ▸ dynamo ▸ forOxford ▸ test ▸ t1 ▸

Current Folder

Name
- ccExample_64
- ccmatrix_ccExample_...
- ccmatrix_ptest
- decoy
- exampleMembrane
- exampleVesicle
- mytest
- ptest
- testClassification_64
- ccExample_64.m
- ccmatrix_ccExample_...
- ccmatrix_ptest.m
- exampleMembrane.ctlg
- exampleVesicle.ctlg
- ptest.m

Details

Workspace

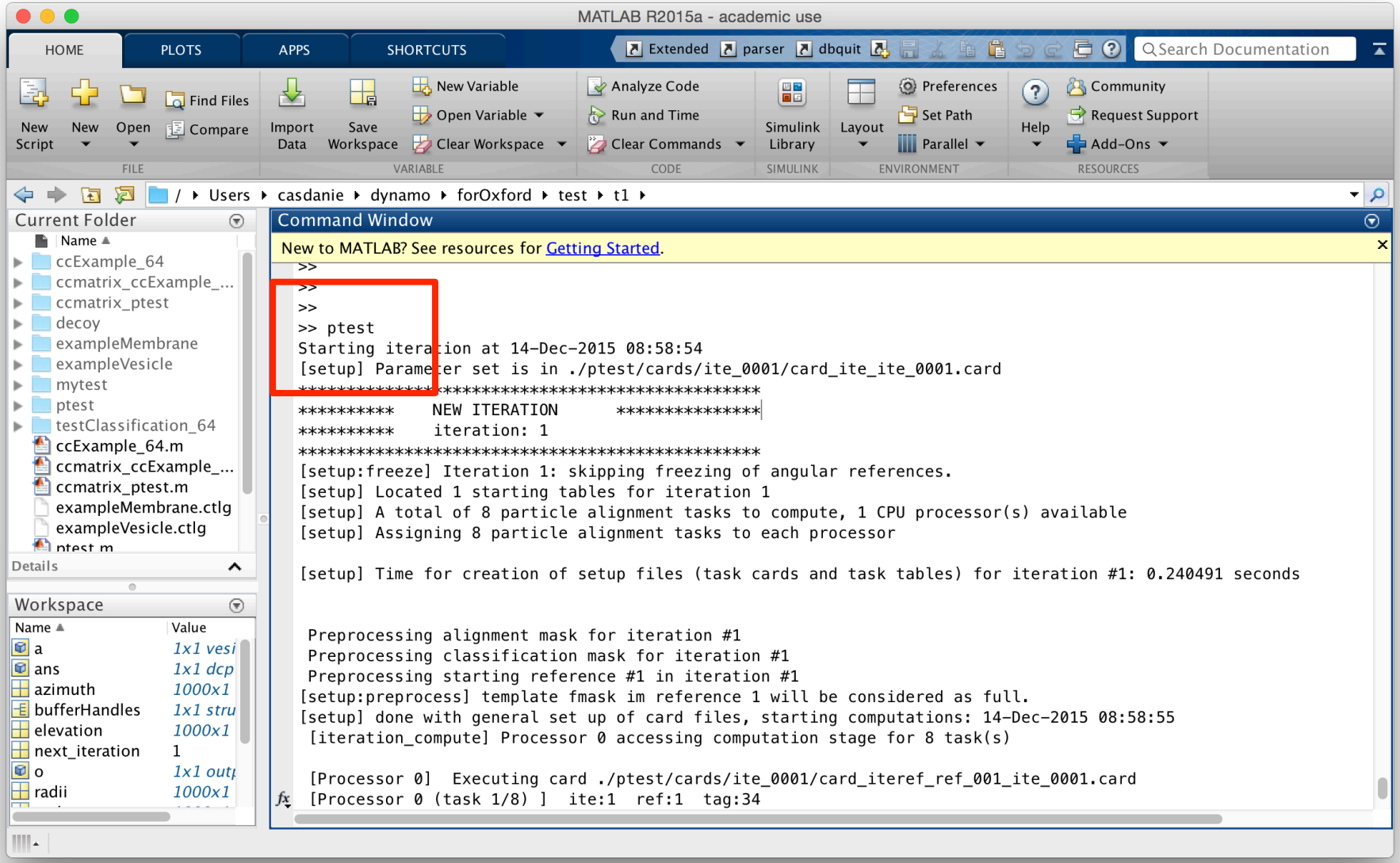
Name	Value
a	1x1 vesi
ans	1x1 dcp
azimuth	1000x1
bufferHandles	1x1 stru
elevation	1000x1
o	1x1 outp
radii	1000x1
rvals	1000x1

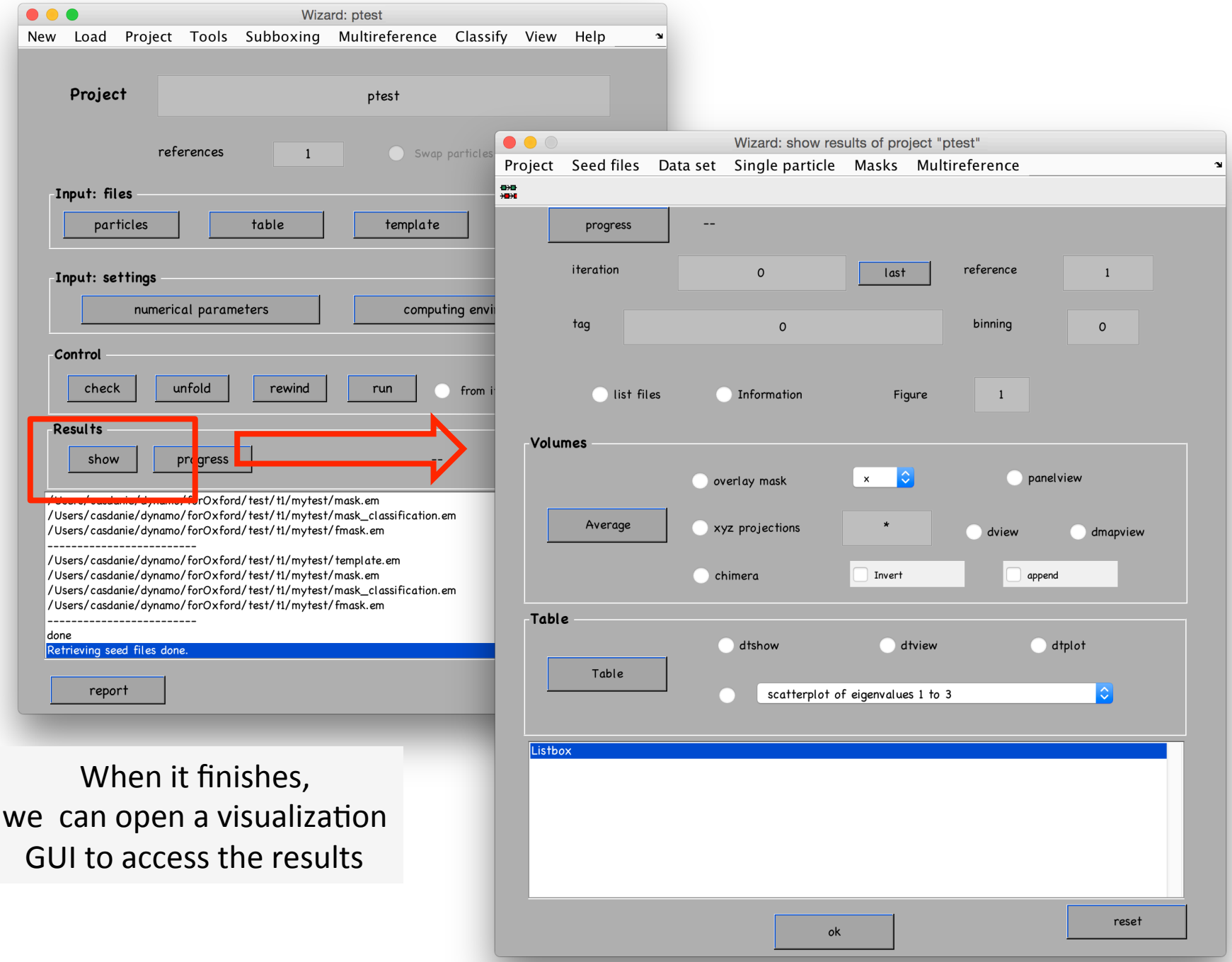
Command Window

New to MATLAB? See resources for Getting Started.

```
Starting iteration at 14-Dec-2015 08:55:55
[setup] Parameter set is in ./ptest/cards/ite_0001/card_ite_ite_0001.card
**************************************************
**********    NEW ITERATION      ***************
**********    iteration: 1
**************************************************
[setup:freeze] Iteration 1: skipping freezing of angular references.
[setup] Located 1 starting tables for iteration 1
[setup] A total of 8 particle alignment tasks to compute, 1 CPU processor(s) available
[setup] Assigning 8 particle alignment tasks to each processor

[setup] Time for creation of setup files (task cards and task tables) for iteration #1: 0.246710 seconds


 Preprocessing alignment mask for iteration #1
 Preprocessing classification mask for iteration #1
 Preprocessing starting reference #1 in iteration #1
[setup:preprocess] template fmask im reference 1 will be considered as full.
[setup] done with general set up of card files, starting computations: 14-Dec-2015 08:55:57
 [iteration_compute] Processor 0 accessing computation stage for 8 task(s)

 [Processor 0]  Executing card ./ptest/cards/ite_0001/card_iteref_ref_001_ite_0001.card
 [Processor 0 (task 1/8) ]  ite:1  ref:1  tag:34
   Coarsest alignment level. Sampling set includes  14 angles with cone_range 360.00 and inplane_range  0.00.
   Processing triplet 1 took  0.03 s. Remaining 13 should take 1s
   Refining level: 1 (18 triplets). Axis: (range 120.00 / step 30.00). Rotation about axis: (range  0.00/ step
```

Note that you can also start a new Matlab/shell and invoke the execution script, what leaves the GUI free to work on other tasks.



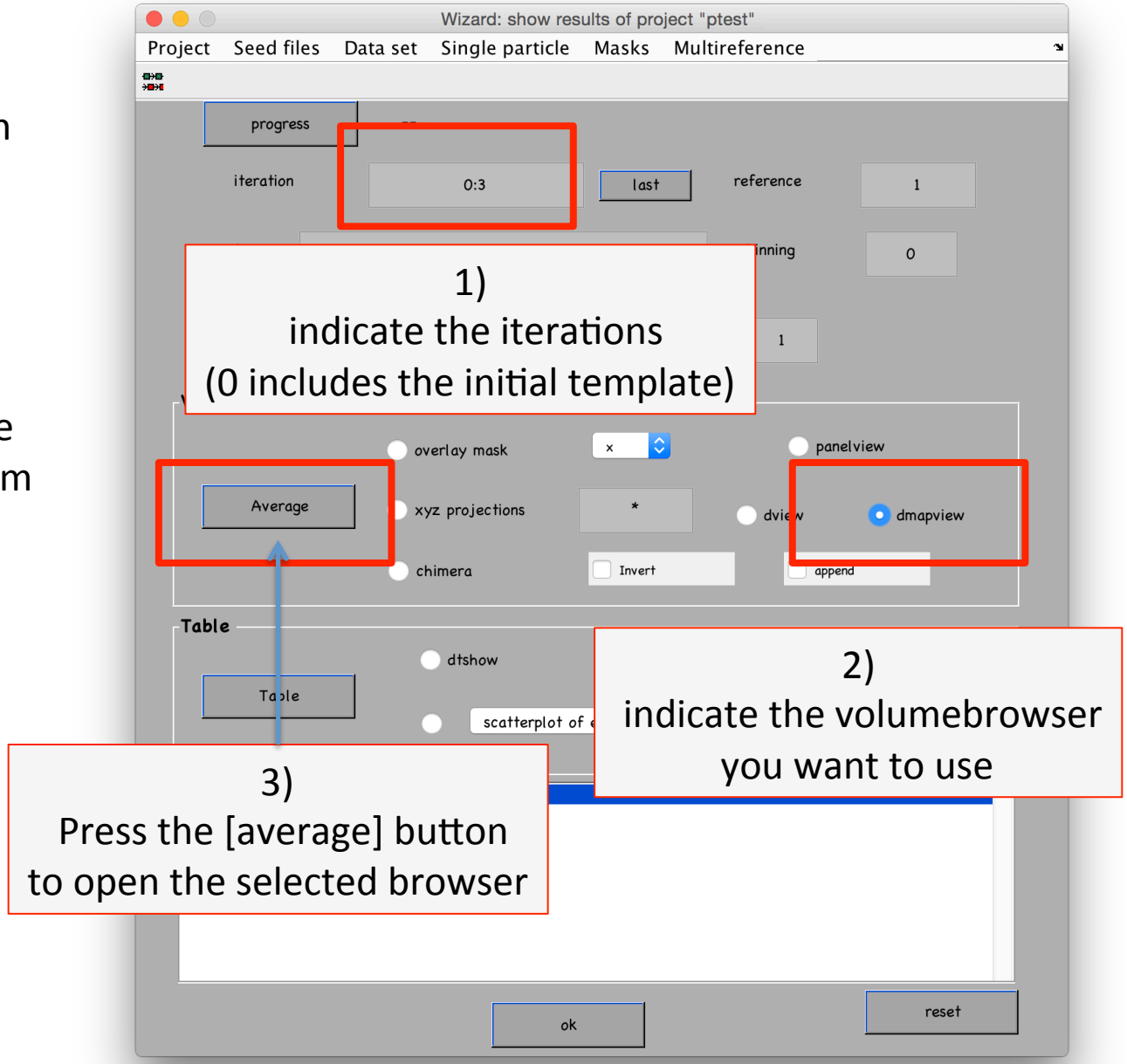


When it finishes,
we can open a visualization
GUI to access the results

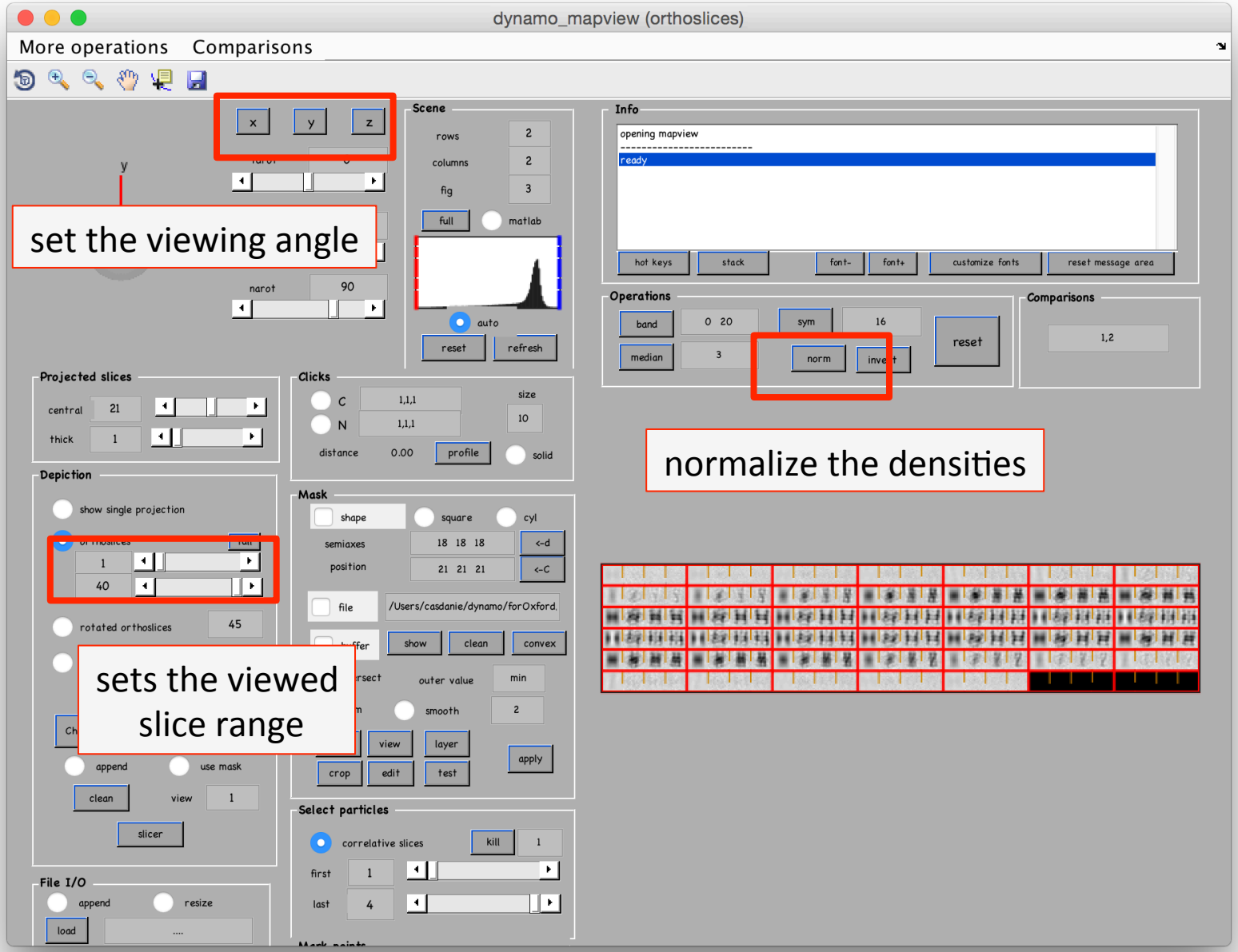
There are many visualization options.

Let us try something basic:

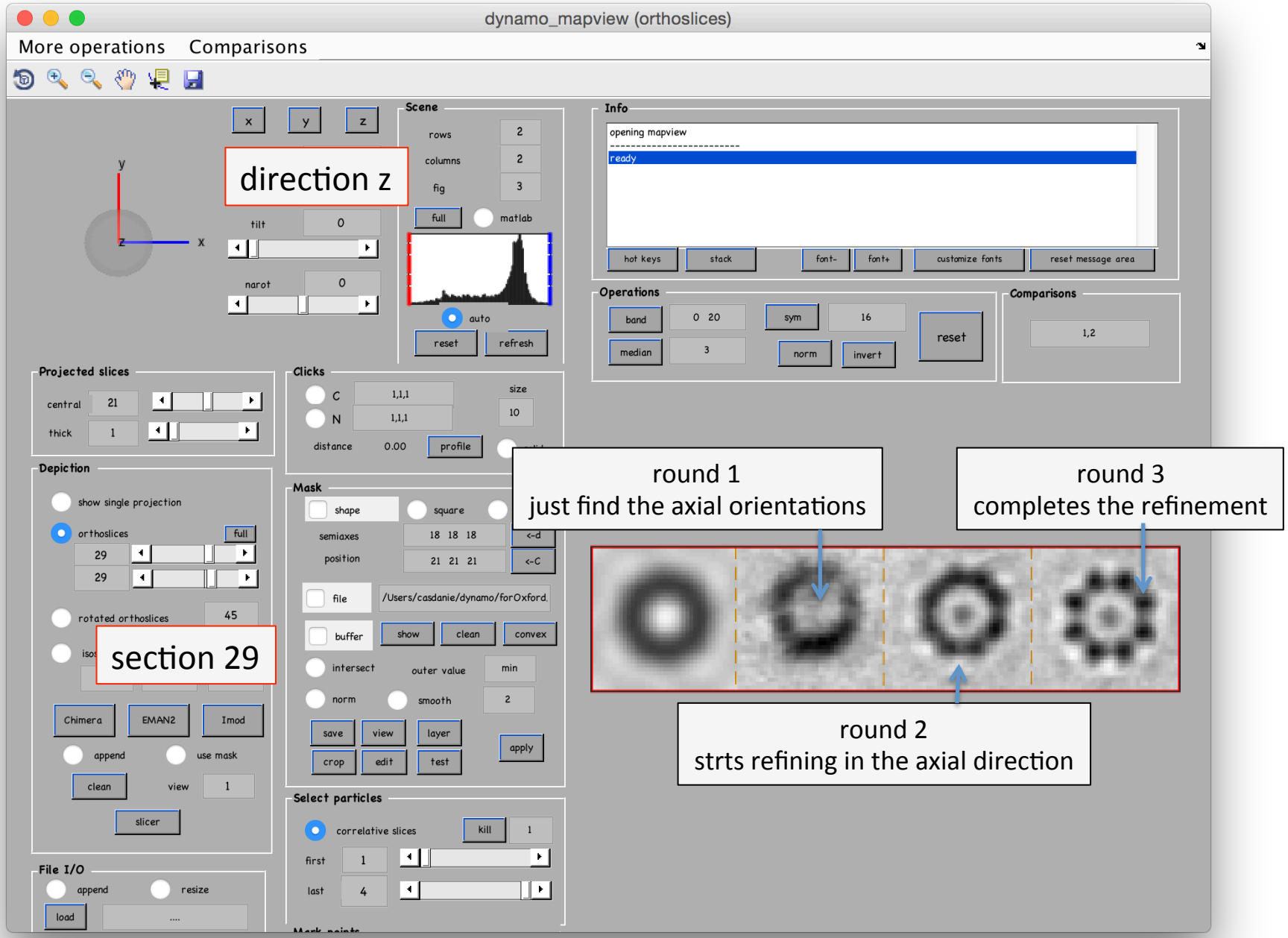
Imagine we want to scan the evolution of the average from iterations 0 to 3



A complicated looking window appears. However, you are interested in a couple of options



For instance, let us choose this visualization setting:



You can also access the created files through the command line using the *Dynamo* database command `ddb`:

picks averages

picks all the iterations

```
>> ddb ptest:a:ite=* -d
/Users/casdanie/dynamo/forOxford/test/t1/mytest/template.em
./ptest/results/ite_0001/averages/average_ref_001_ite_0001.em
./ptest/results/ite_0002/averages/average_ref_001_ite_0002.em
./ptest/results/ite_0003/averages/average_ref_001_ite_0003.em
>>
```

just display the file names on screen

`ddb ptest:a:ite=* -m`
would send the results to `mapview` directly

The files can then be opened with `mapview, dview`, etc
You can also access tables (`rt`), data (`data`), templates, etc....
Type doc ddb for the syntax

Exercise

Create a tutorial data set without generating automatically a project:

```
>> dtutorial newtest
```

… and create and run a project from scratch to align the particles