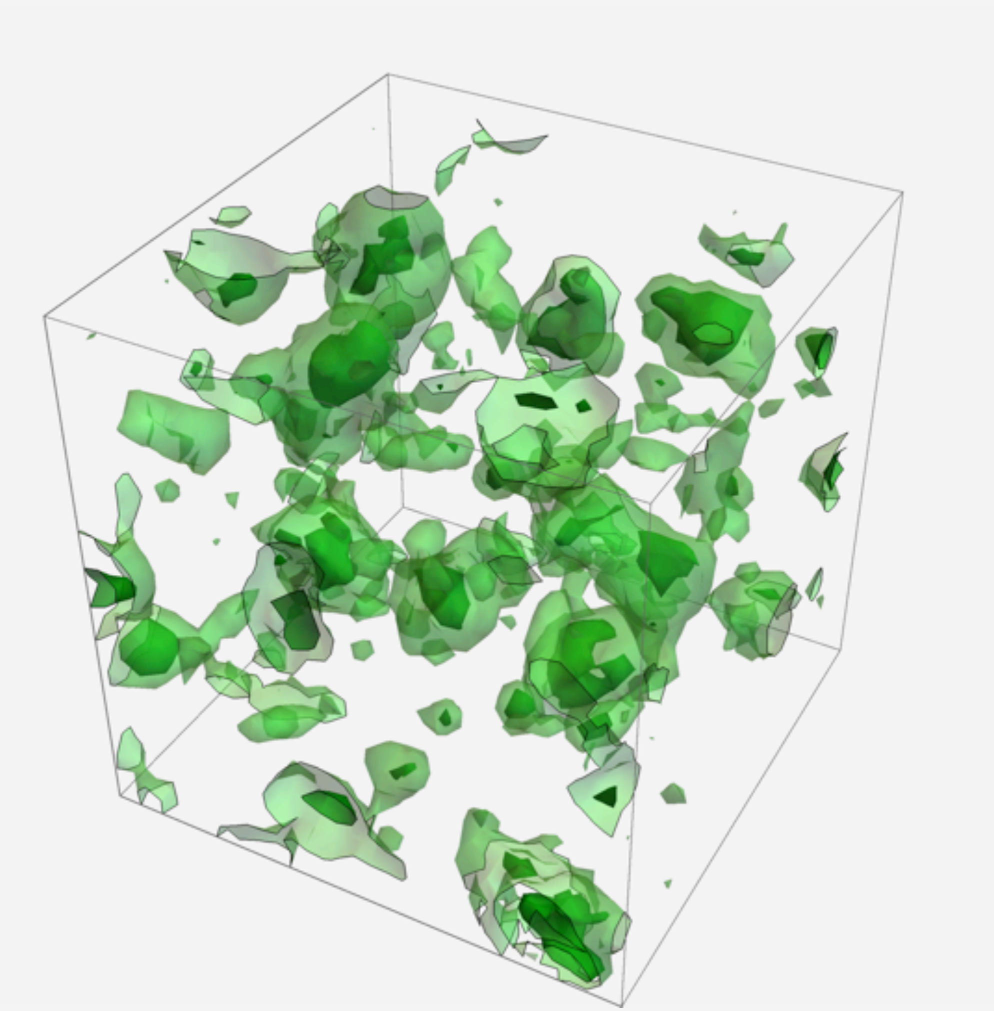
𝒞osmoℒattice School:

Lecture 10: Plotting 3D Data with 𝒞osmoℒattice

Kenneth Marschall

Goal of the lecture

Run a simulation and print snapshots of the energy distributions

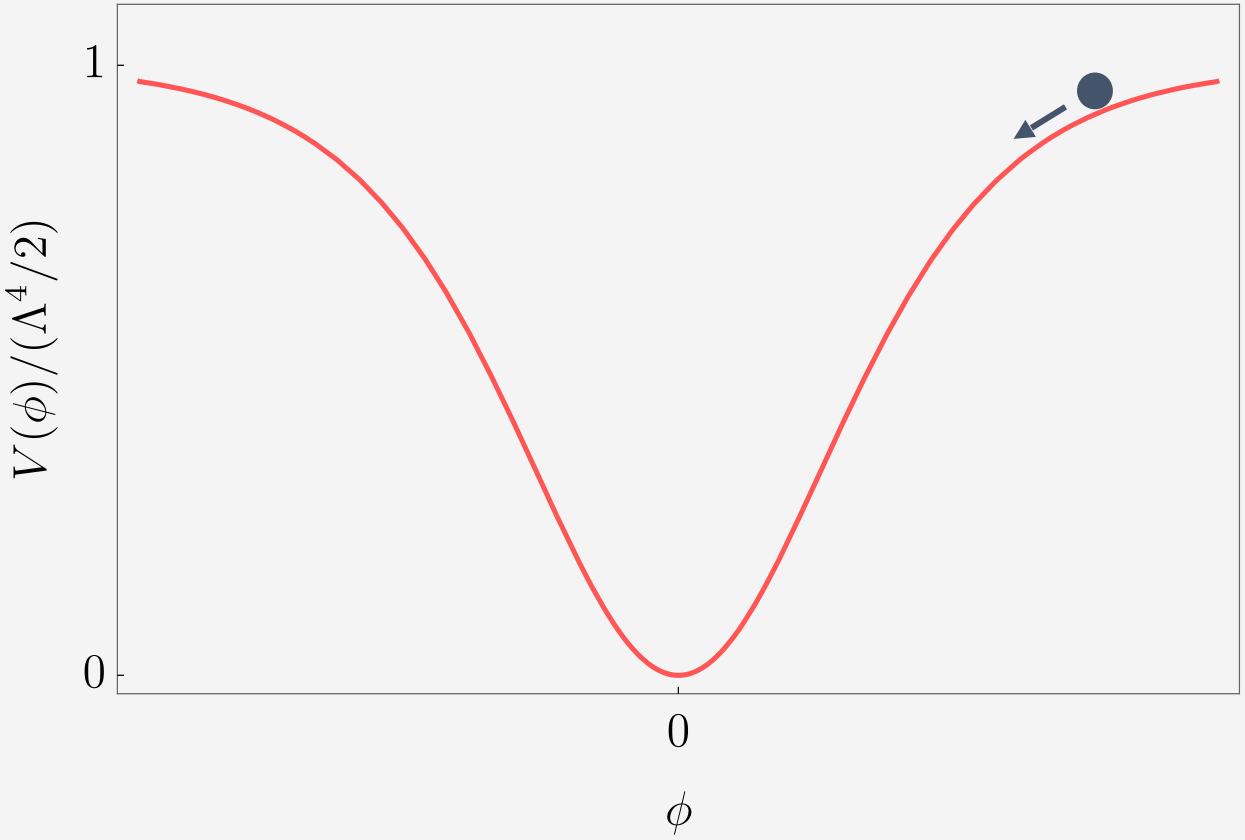


Motivation

Consider the potential:

$$V(\phi, X) = \frac{1}{2}\Lambda^4 \tanh^2\left(\frac{\phi}{M}\right) + \frac{1}{2}g^2\phi^2X^2$$

with $M = 0.01 m_{\mathrm{pl}}$

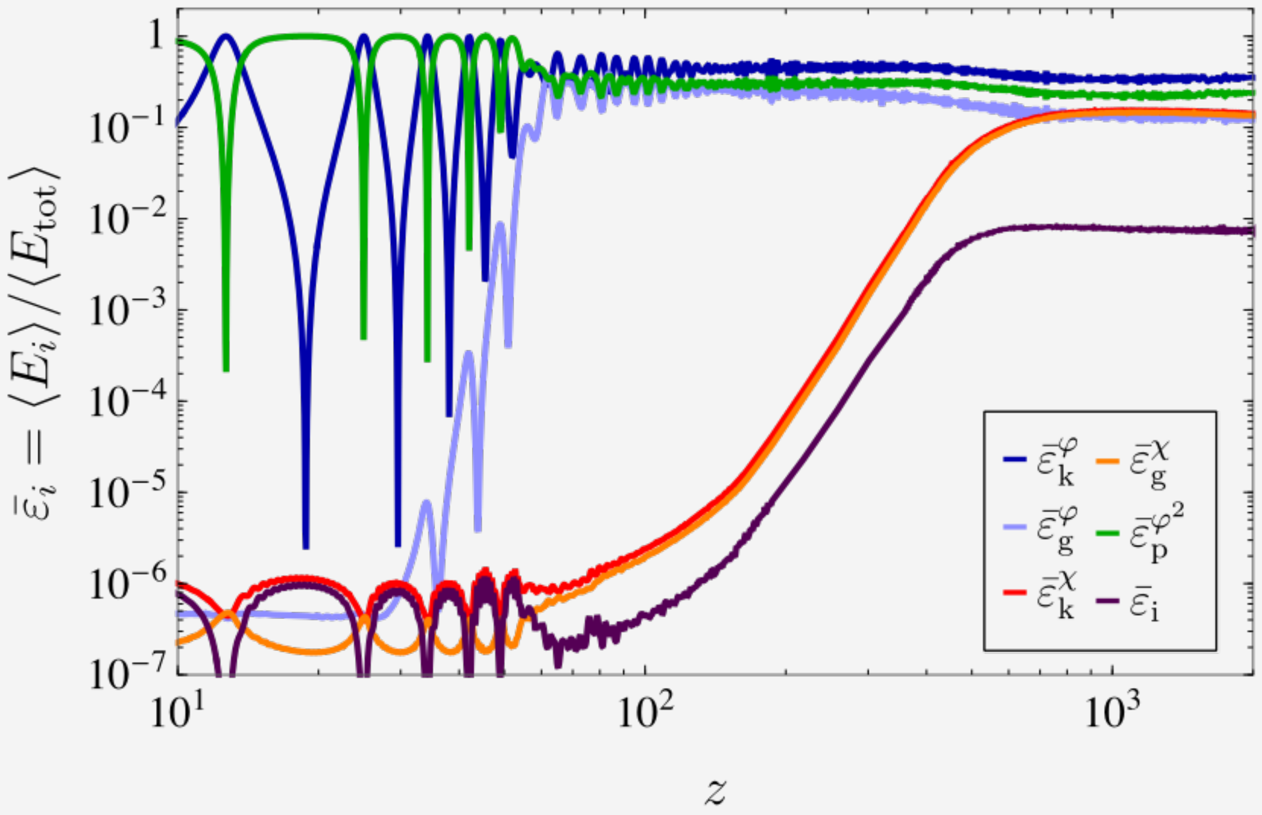


Motivation

Consider the potential:

$$V(\phi, X) = \frac{1}{2}\Lambda^4 \tanh^2\left(\frac{\phi}{M}\right) + \frac{1}{2}g^2\phi^2X^2$$

with $M = 0.01 m_{\rm pl}$

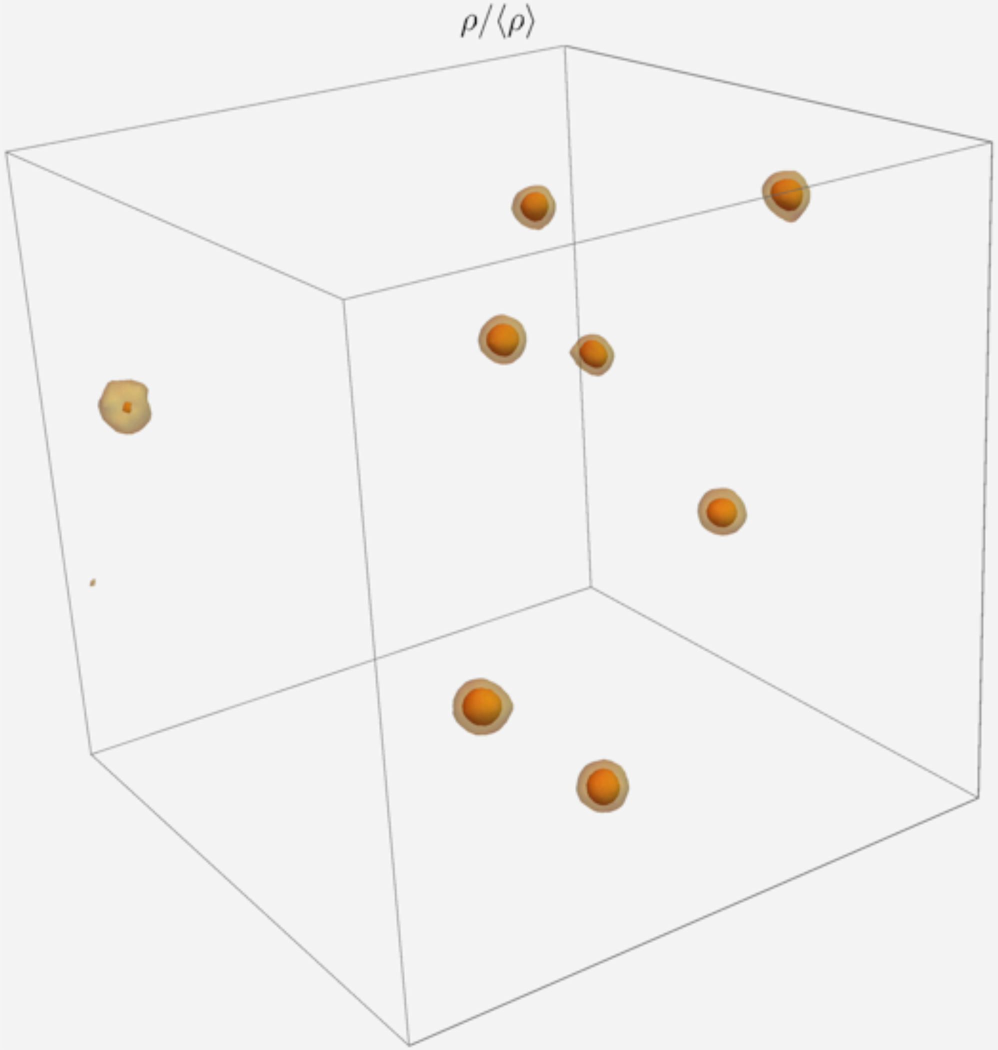


Motivation

Consider the potential:

$$V(\phi, X) = \frac{1}{2}\Lambda^4 \tanh^2\left(\frac{\phi}{M}\right) + \frac{1}{2}g^2\phi^2X^2$$

with $M = 0.01 m_{\rm pl}$

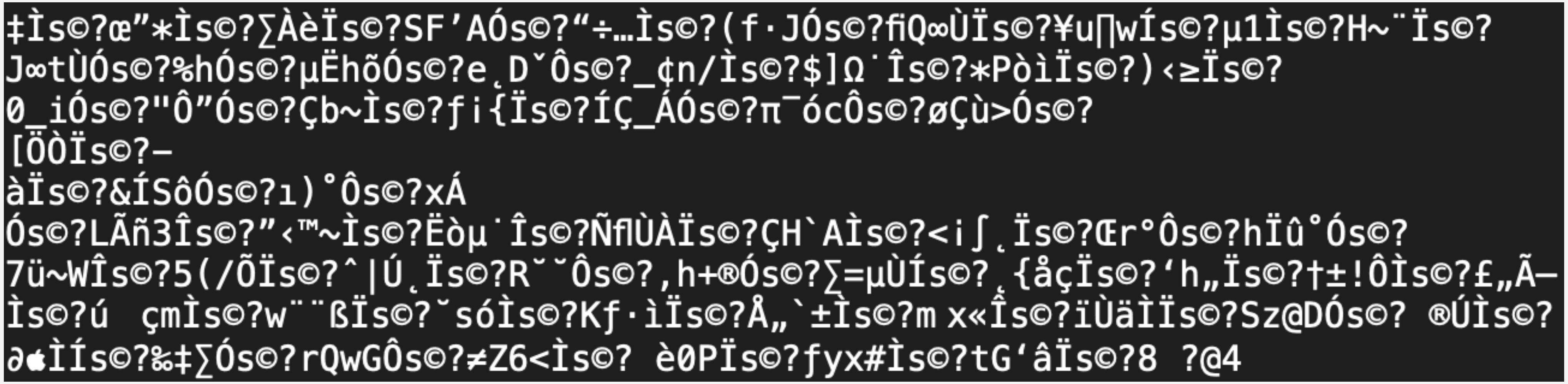


Hierarchical Data Format (HDF5)

CosmoLattice uses the HDF5 library to save snapshots of the three dimensional distributions of the different energy components of the simulated system.

Hierarchical Data Format (HDF5)

If you open an HDF5 file in e.g. a text editor, you’ll find weird stuff:



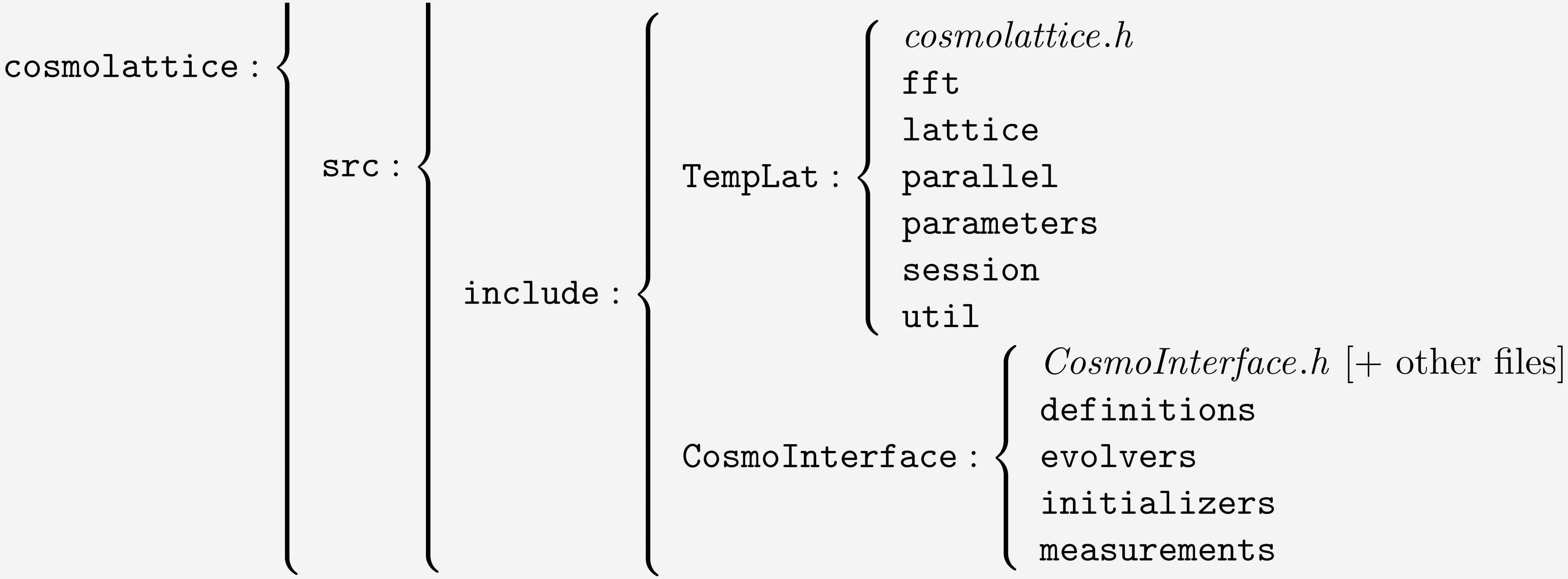
```
‡Ìs©?œ”*Ìs©?∑ÀèÏs©?SF’AÓs©?“÷…Ìs©?(f·JÓs©?ﬁQ∞ÙÏs©?¥u∏wÍs©?µ1Ìs©?H~¨Ïs©?
J∞tÙÓs©?%hÓs©?µËhõÓs©?e˛DˇÔs©?_¢n/Ìs©?$]Ω˙Îs©?*PòìÏs©?)‹≥Ïs©?
0_iÓs©?"Ô”Ós©?Çb~Ìs©?ƒ¡{Ïs©?ÍÇ_ÁÓs©?π¯ócÔs©?øÇù>Ós©?
[ÖÒÏs©?–
àÏs©?&ÍSôÓs©?ı)˚Ôs©?xÁ
Ós©?LÃñ3Îs©?”‹™~Ìs©?ËòµÎs©?ÑﬂÙÀÏs©?ÇH`AÌs©?<i∫˛Ïs©?Œr°Ôs©?hÏû˚Ós©?
7ü~WÎs©?5(/ÕÏs©?ˆ|Ú˛Ïs©?R˘˘Ôs©?,h+®Ós©?∑=µÙÍs©?˛{åçÏs©?‘h„Ïs©?†±!ÔÌs©?£„Ã–
Ìs©?ú çmÌs©?w¨¨ßÏs©?˘sóÌs©?Kƒ·ìÏs©?Å„`±Ìs©?m x«Îs©?ïÙäÌÏs©?Sz@DÓs©? ®ÚÌs©?
∂ÌÍs©?‰‡∑Ós©?rQwGÔs©?≠Z6<Ìs©? è0PÏs©?ƒyx#Ìs©?tG‘âÏs©?8 ?@4
```

To read .h5 files use: Mathematica, Matlab, python, gnuplot, etc.

Structure of HDF5 files in CosmoLattice

Snapshots of the different energy distributions are taken by:

energysnapshotmeasurer.h

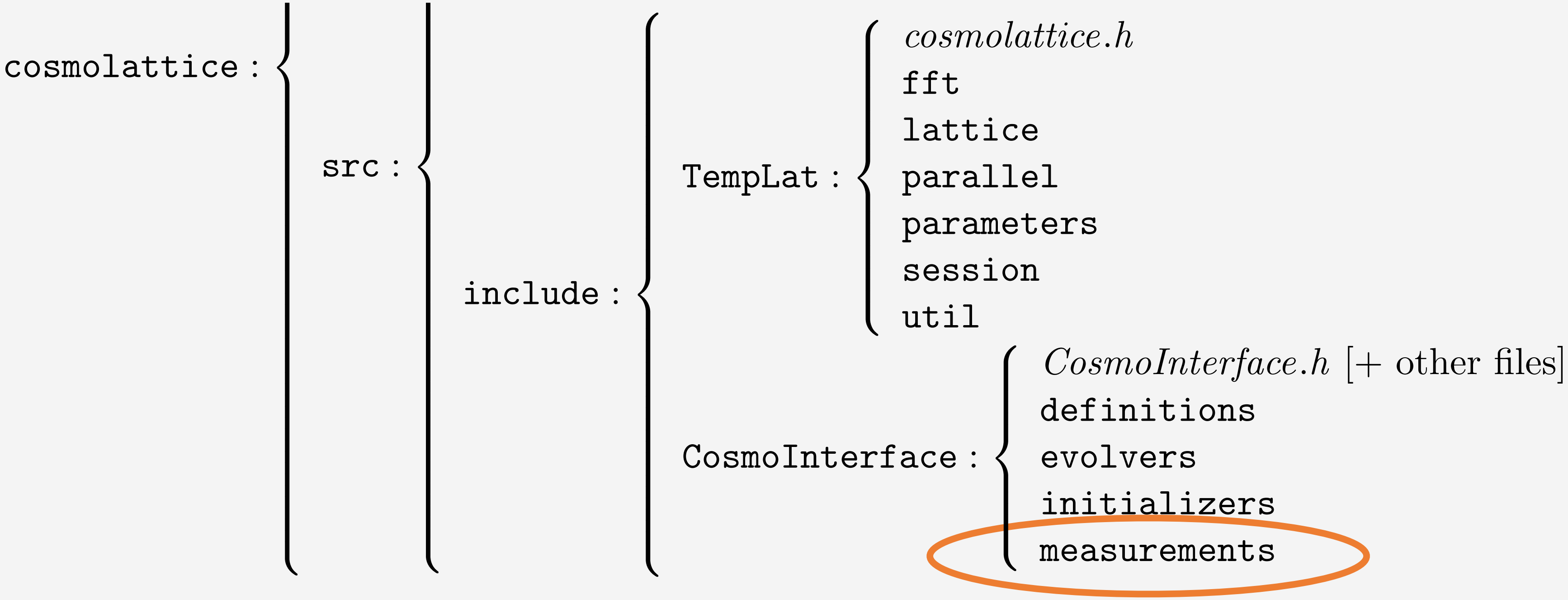


- `cosmolattice` :
  - `src` :
    - `include` :
      - `TempLat` :
        - *cosmolattice.h*
        - `fft`
        - `lattice`
        - `parallel`
        - `parameters`
        - `session`
        - `util`
      - `CosmoInterface` :
        - *CosmoInterface.h* [+ other files]
        - `definitions`
        - `evolvers`
        - `initializers`
        - `measurements`

Structure of HDF5 files in CosmoLattice

Snapshots of the different energy distributions are taken by:

energysnapshotmeasurer.h



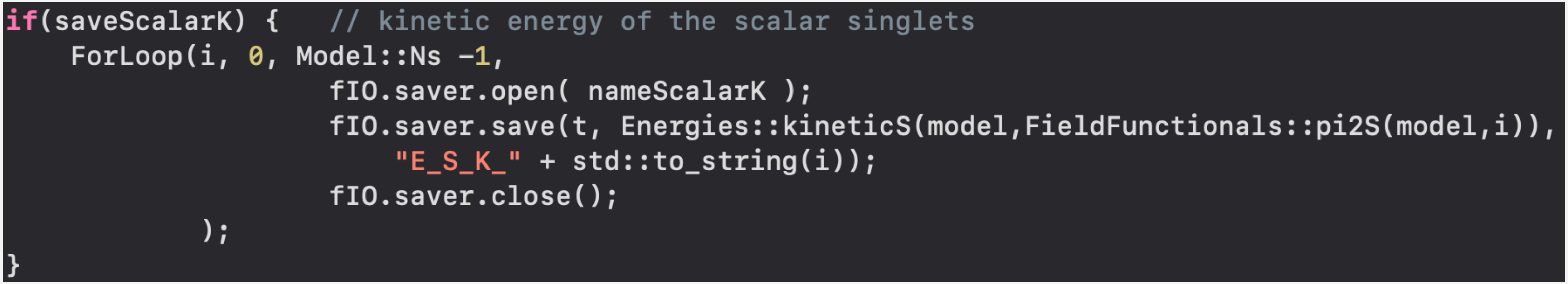
- cosmolattice :
  - src :
    - include :
      - TempLat :
        - *cosmolattice.h*
        - fft
        - lattice
        - parallel
        - parameters
        - session
        - util
      - CosmoInterface :
        - *CosmoInterface.h* [+ other files]
        - definitions
        - evolvers
        - initializers
        - measurements

Structure of HDF5 files in CosmoLattice

Snapshots of the different energy distributions are taken by:

energysnapshotmeasurer.h

Snapshots of e.g. the kinetic energy density of a scalar field is ‘taken‘ by:



```
if(saveScalarK) {   // kinetic energy of the scalar singlets
    ForLoop(i, 0, Model::Ns -1,
                    fIO.saver.open( nameScalarK );
                    fIO.saver.save(t, Energies::kineticS(model,FieldFunctionals::pi2S(model,i)),
                        "E_S_K_" + std::to_string(i));
                    fIO.saver.close();
            );
}
```

Structure of HDF5 files in CosmoLattice

CL saves the different energy components in .h5 files with the name structure:

type_energy_snapshot_fieldtype.h5

	type	fieldtype
Scalars	kinetic, gradient	scalar, complex_scalar, SU2_doublet
Gauge fields	electric, magnetic	U1, SU2

The potential energy is saved in: potential_energy_snapshot.h5

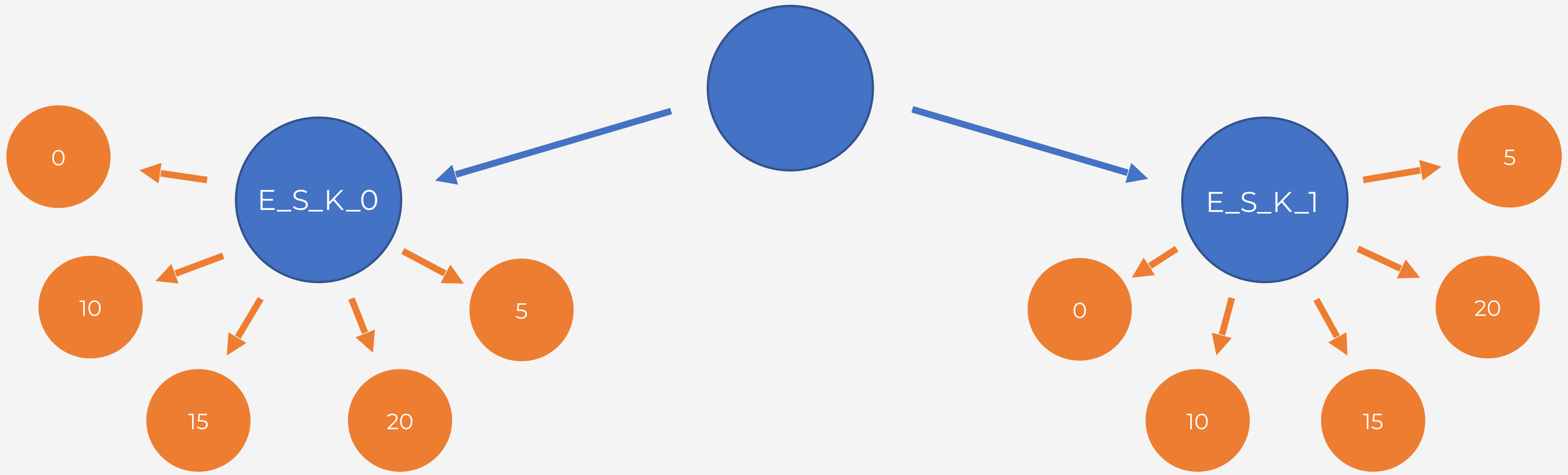
Structure of HDF5 files in CosmoLattice

HDF5 files are structured in Groups and Datasets.

Structure of HDF5 files in CosmoLattice

HDF5 files are structured in Groups and Datasets.

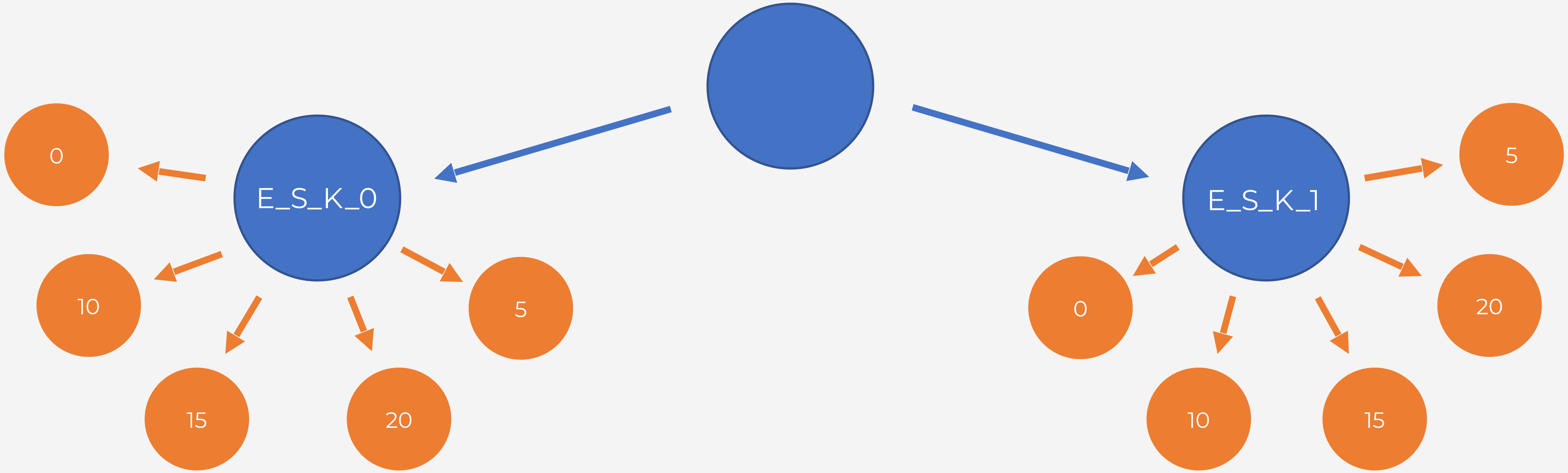
Example: The kinetic_energy_snapshot_scalar.h5 file of a simulation with two real scalar fields is structured as follows



Structure of HDF5 files in CosmoLattice

HDF5 files are structured in Groups and Datasets.

Example: The kinetic_energy_snapshot_scalar.h5 file of a simulation with two real scalar fields is structured as follows



In Mathematica:

```
Import["kinetic_energy_snapshot_scalar.h5", "Summary"]
Import["kinetic_energy_snapshot_scalar.h5", "StructureGraph"]
```

Structure of HDF5 files in CosmoLattice

The Groups of the different fields are named as follows:

Scalar fields:	kinetic	gradient
Scalar singlet	E_S_K_#	E_S_G_#
Complex scalar	E_CS_K_#	E_CS_G_#
SU(2) doublet	E_SU2D_K_#	E_SU2D_G_#

Gauge fields:	electric	magnetic
U(1)	E_A_K_#	E_A_G_#
SU(2)	E_B_K_#	E_B_G_#

Structure of HDF5 files in CosmoLattice

We consider a simulation with $N_{\mathrm{grid}} = 4$. Snapshots can be imported into e.g. Mathematica by:

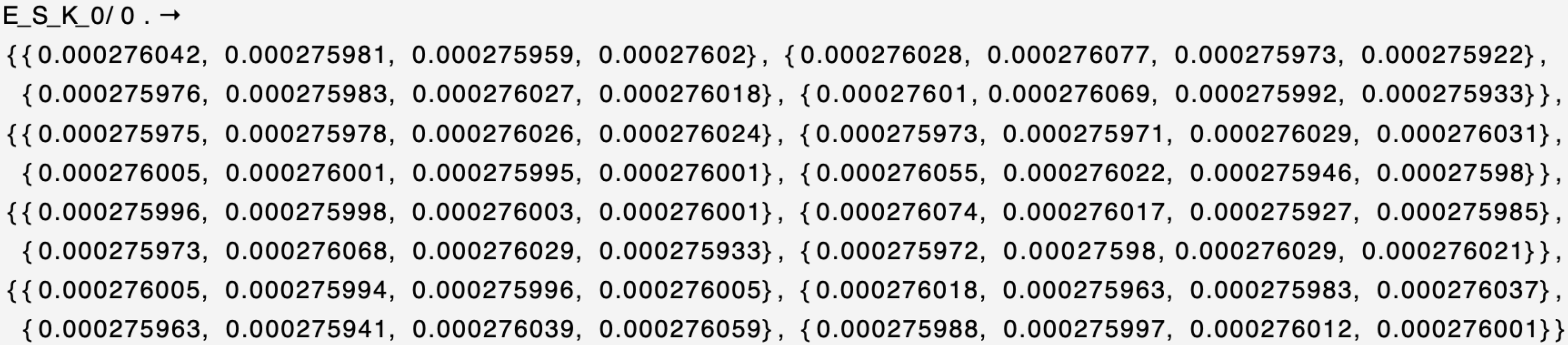
```
Import["kinetic_energy_snapshot_scalar.h5", {"Data"}]
```

Structure of HDF5 files in CosmoLattice

We consider a simulation with $N_{\rm grid} = 4$. Snapshots can be imported into e.g. Mathematica by:

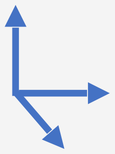
```
Import["kinetic_energy_snapshot_scalar.h5", {"Data"}]
```

The first kinetic energy snapshot of the first scalar field is structured as follows:

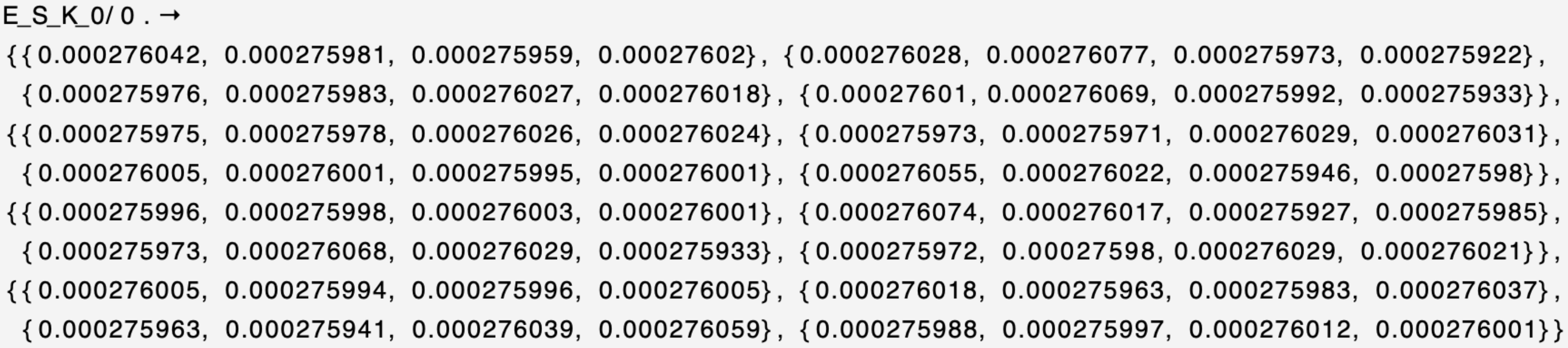


```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice

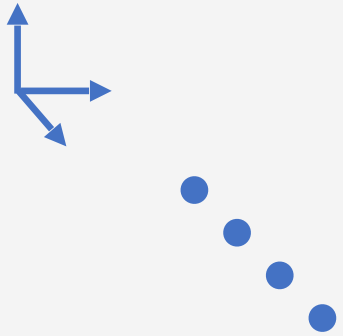


The first kinetic energy snapshot of the first scalar field is structured as follows:

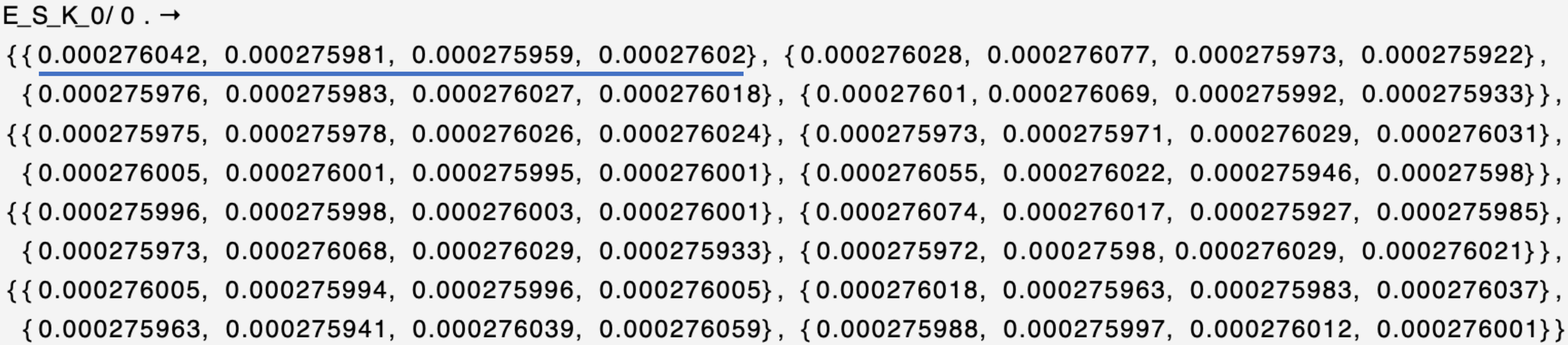


```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice

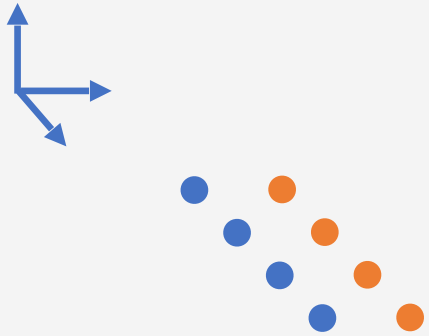


The first kinetic energy snapshot of the first scalar field is structured as follows:

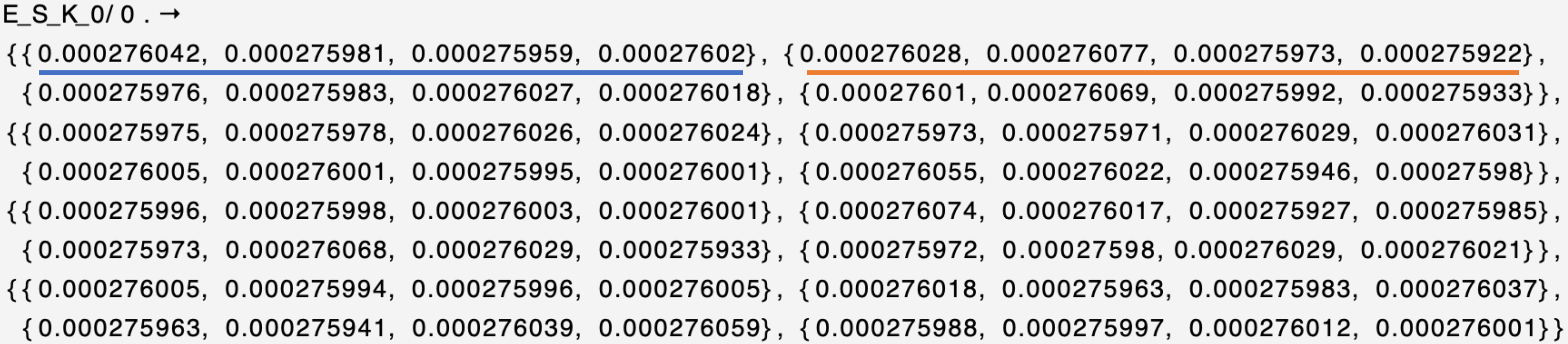


```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice

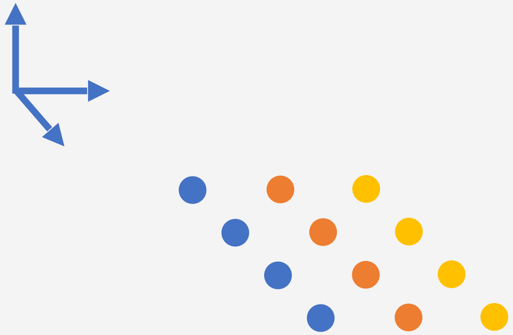


The first kinetic energy snapshot of the first scalar field is structured as follows:

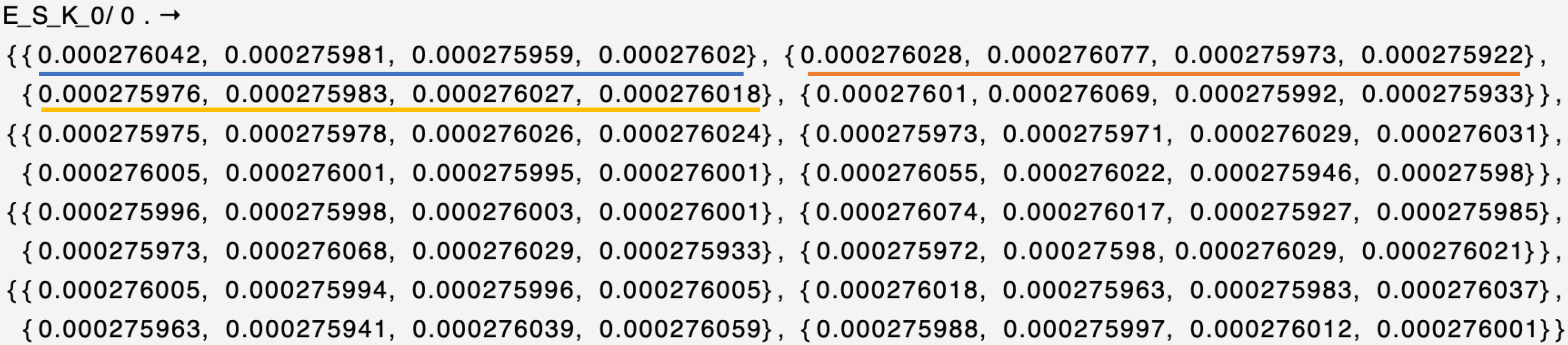


```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
  {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
  {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
  {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
  {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice

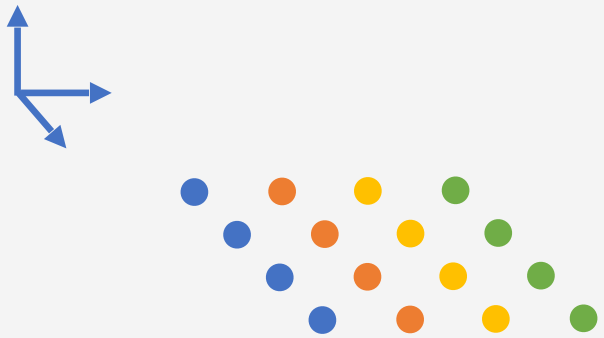


The first kinetic energy snapshot of the first scalar field is structured as follows:



```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

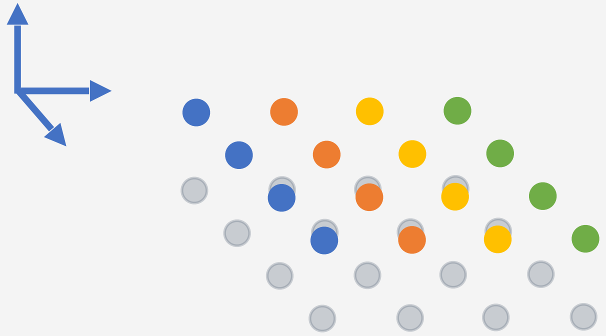
Structure of HDF5 files in CosmoLattice



The first kinetic energy snapshot of the first scalar field is structured as follows:

```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

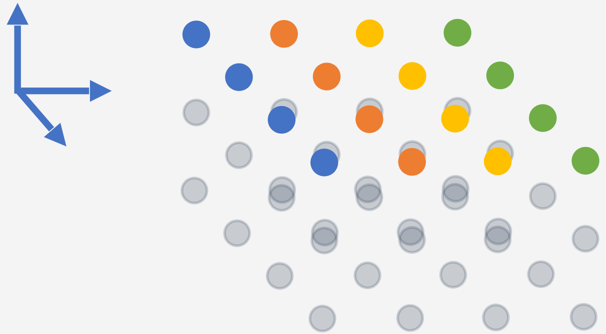
Structure of HDF5 files in CosmoLattice



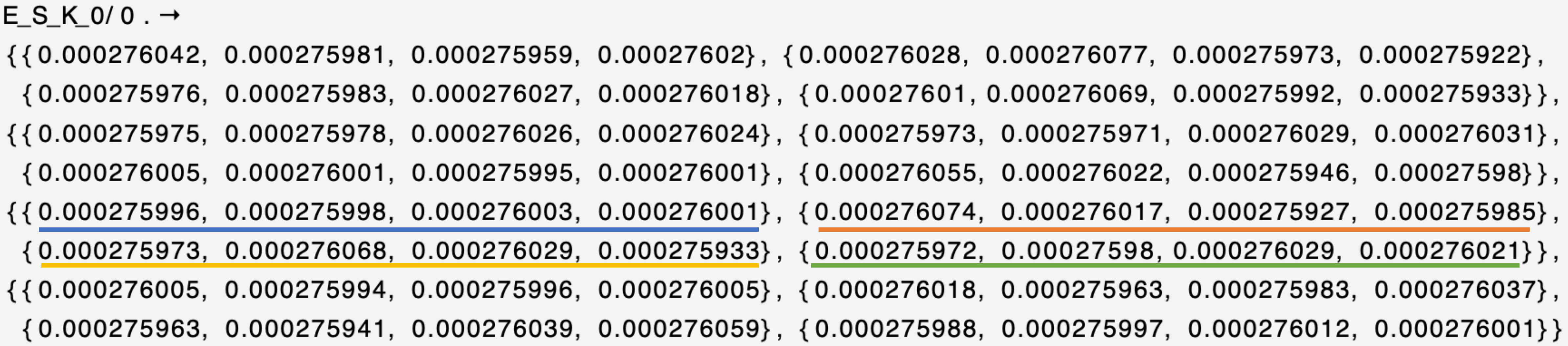
The first kinetic energy snapshot of the first scalar field is structured as follows:

```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice

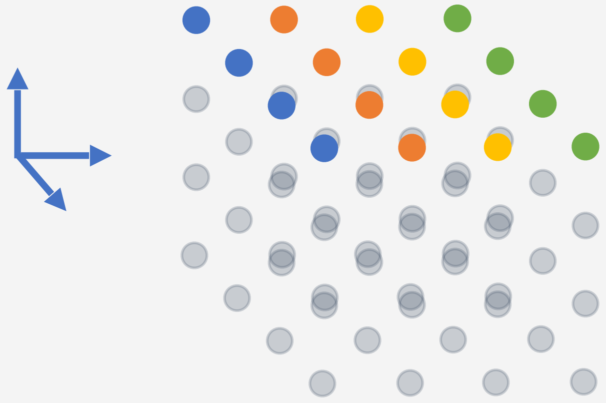


The first kinetic energy snapshot of the first scalar field is structured as follows:

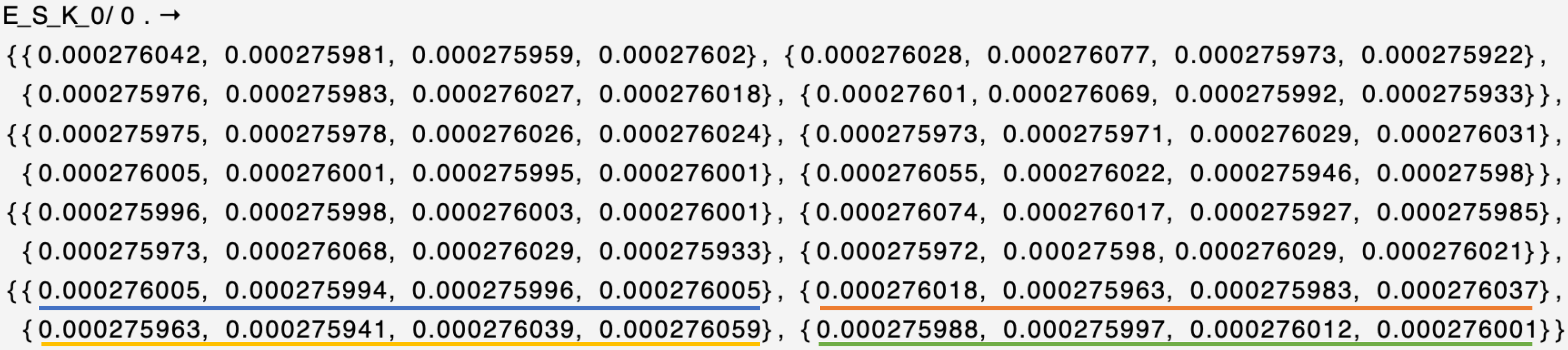


```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice



The first kinetic energy snapshot of the first scalar field is structured as follows:



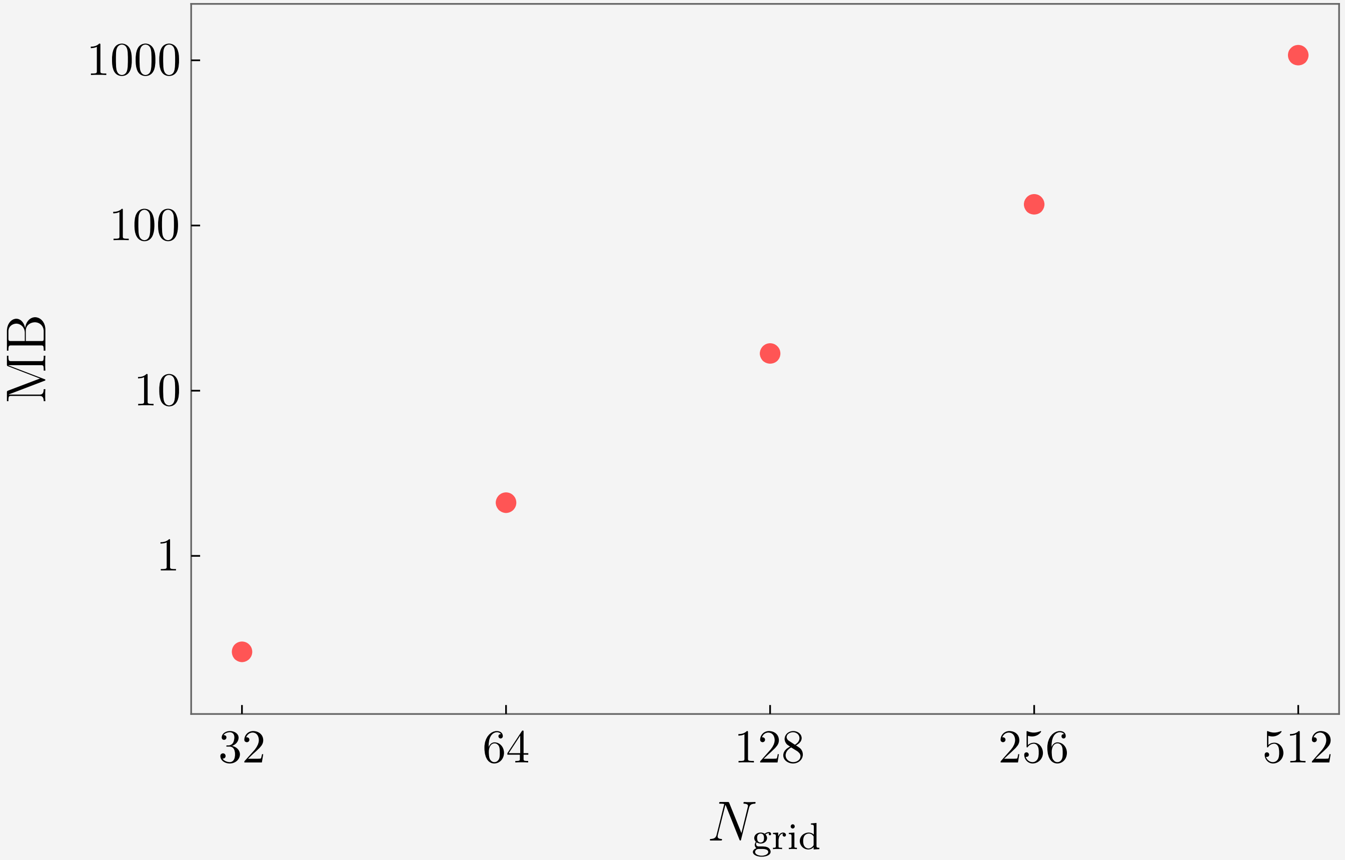
```
E_S_K_0/ 0 . →
{{0.000276042, 0.000275981, 0.000275959, 0.00027602}, {0.000276028, 0.000276077, 0.000275973, 0.000275922},
 {0.000275976, 0.000275983, 0.000276027, 0.000276018}, {0.00027601, 0.000276069, 0.000275992, 0.000275933}},
{{0.000275975, 0.000275978, 0.000276026, 0.000276024}, {0.000275973, 0.000275971, 0.000276029, 0.000276031},
 {0.000276005, 0.000276001, 0.000275995, 0.000276001}, {0.000276055, 0.000276022, 0.000275946, 0.00027598}},
{{0.000275996, 0.000275998, 0.000276003, 0.000276001}, {0.000276074, 0.000276017, 0.000275927, 0.000275985},
 {0.000275973, 0.000276068, 0.000276029, 0.000275933}, {0.000275972, 0.00027598, 0.000276029, 0.000276021}},
{{0.000276005, 0.000275994, 0.000275996, 0.000276005}, {0.000276018, 0.000275963, 0.000275983, 0.000276037},
 {0.000275963, 0.000275941, 0.000276039, 0.000276059}, {0.000275988, 0.000275997, 0.000276012, 0.000276001}}
```

Structure of HDF5 files in CosmoLattice

A word of caution:

one slice of a simulation of one field with $N_{\rm grid} = 128$ requires $\sim 17\,\mathrm{MB}$!

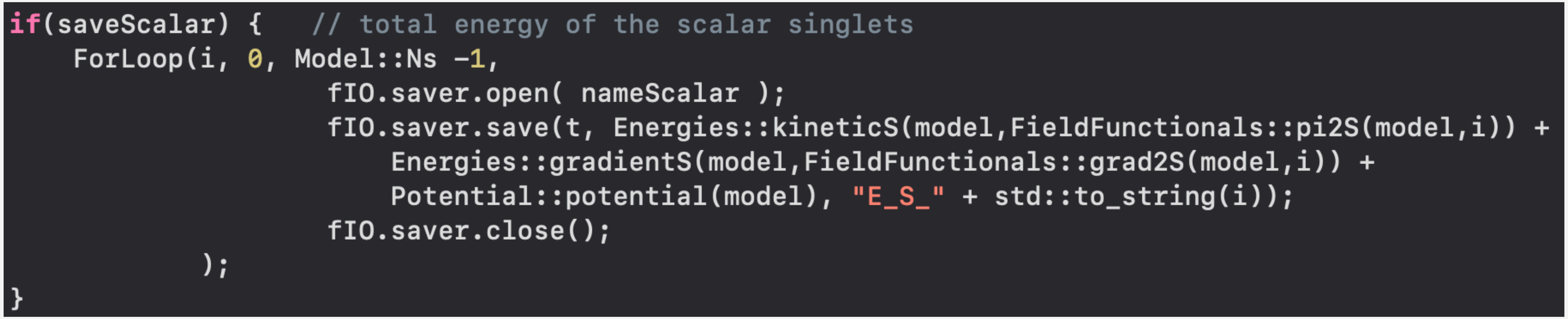
For a simulation with $N_{\rm grid} = 256$ you need $8\times$ more storage, etc.



Structure of h5 files in CosmoLattice

A word of caution:

If your interested in other quantities, e.g. the total energy density of a scalar singlet, you can add it in the energysnapshotmeasurer.h to save storage:

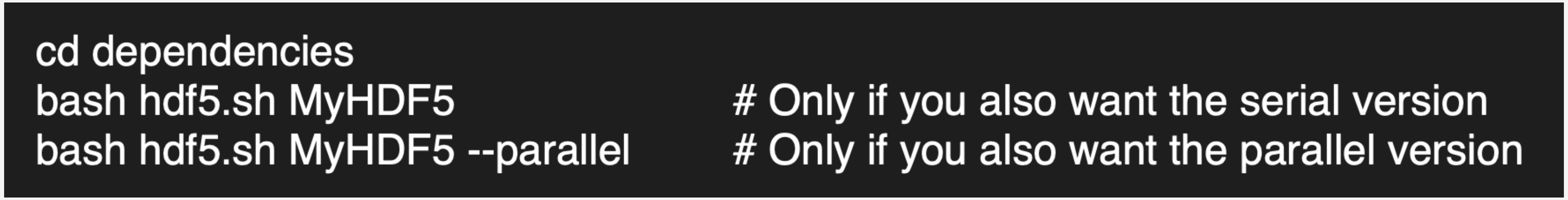


```
if(saveScalar) {   // total energy of the scalar singlets
    ForLoop(i, 0, Model::Ns -1,
                    fIO.saver.open( nameScalar );
                    fIO.saver.save(t, Energies::kineticS(model,FieldFunctionals::pi2S(model,i)) +
                        Energies::gradientS(model,FieldFunctionals::grad2S(model,i)) +
                        Potential::potential(model), "E_S_" + std::to_string(i));
                    fIO.saver.close();
            );
}
```

Examples

Preparation: HDF5

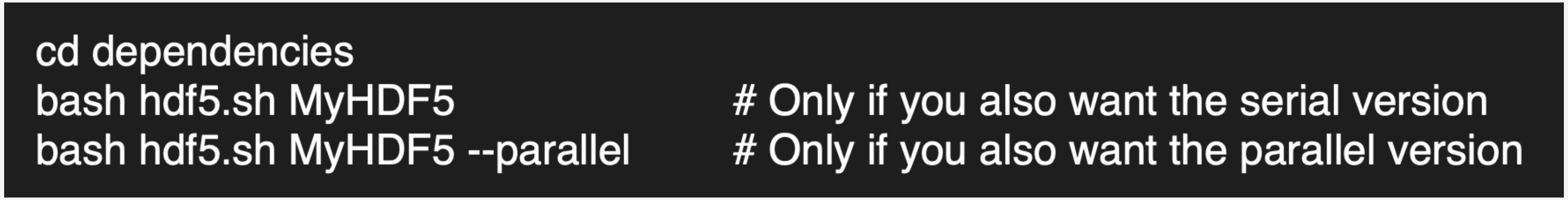
First, we have to make our computers fit for snapshot printing. To do so, open your Terminal and go to the cosmolattice folder. Then, install the HDF5 library by:



```
cd dependencies
bash hdf5.sh MyHDF5                   # Only if you also want the serial version
bash hdf5.sh MyHDF5 --parallel        # Only if you also want the parallel version
```

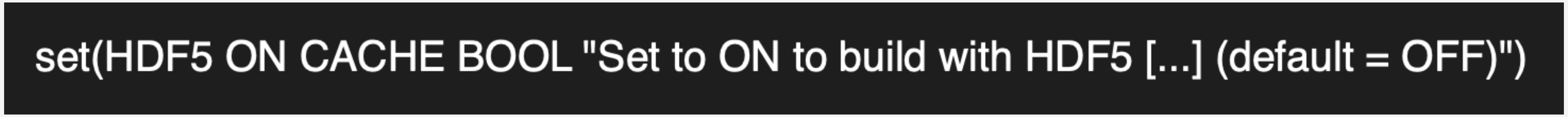
Preparation: HDF5

First, we have to make our computers fit for snapshot printing. To do so, open your Terminal and go to the cosmolattice folder. Then, install the HDF5 library by:



```
cd dependencies
bash hdf5.sh MyHDF5                  # Only if you also want the serial version
bash hdf5.sh MyHDF5 --parallel       # Only if you also want the parallel version
```

Then, in CMakeLists.txt we have to set HDF5 from ’OFF’ to ‘ON’ (line 15):

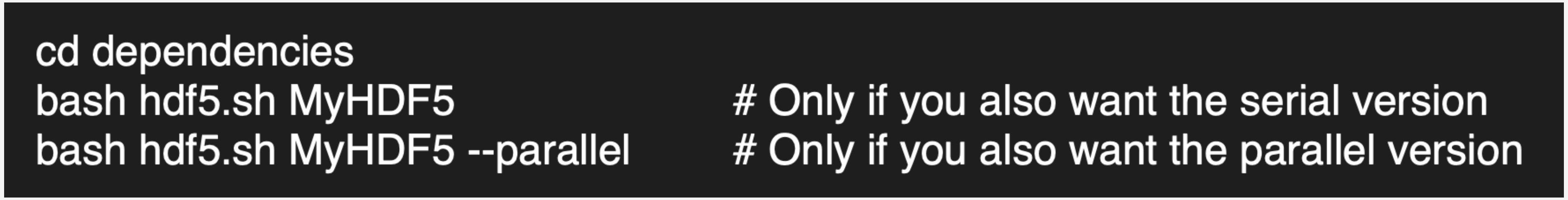


```
set(HDF5 ON CACHE BOOL "Set to ON to build with HDF5 [...] (default = OFF)")
```

(Also, you might need to include the MYHDF5_PATH (line 56) in CMakeLists.txt)

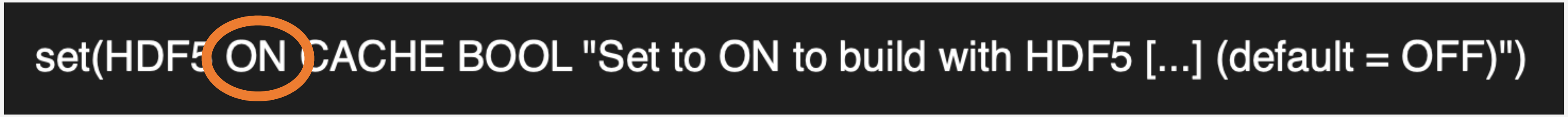
Preparation: HDF5

First, we have to make our computers fit for snapshot printing. To do so, open your Terminal and go to the cosmolattice folder. Then, install the HDF5 library by:



```
cd dependencies
bash hdf5.sh MyHDF5                # Only if you also want the serial version
bash hdf5.sh MyHDF5 --parallel     # Only if you also want the parallel version
```

Then, in CMakeLists.txt we have to set HDF5 from ’OFF’ to ‘ON’ (line 15):

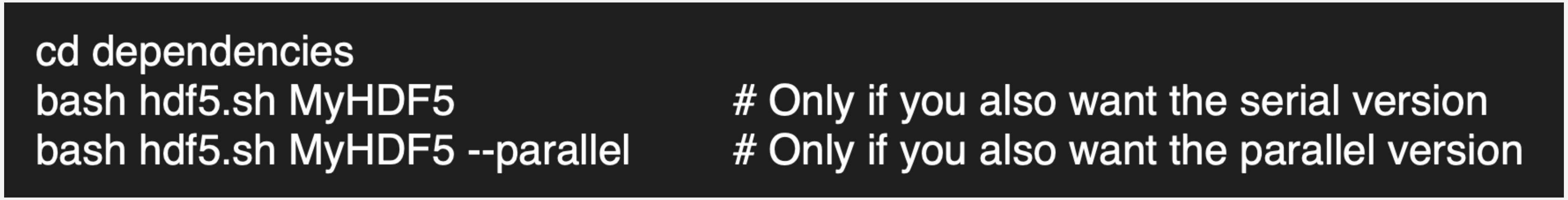


```
set(HDF5 ON CACHE BOOL "Set to ON to build with HDF5 [...] (default = OFF)")
```

(Also, you might need to include the MYHDF5_PATH (line 56) in CMakeLists.txt)

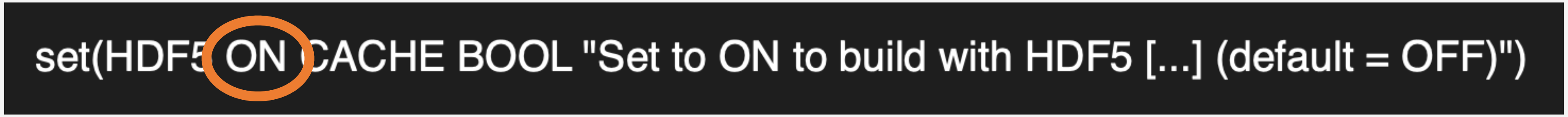
Preparation: HDF5

First, we have to make our computers fit for snapshot printing. To do so, open your Terminal and go to the cosmolattice folder. Then, install the HDF5 library by:



```
cd dependencies
bash hdf5.sh MyHDF5                 # Only if you also want the serial version
bash hdf5.sh MyHDF5 --parallel      # Only if you also want the parallel version
```

Then, in CMakeLists.txt we have to set HDF5 from ’OFF’ to ‘ON’ (line 15):

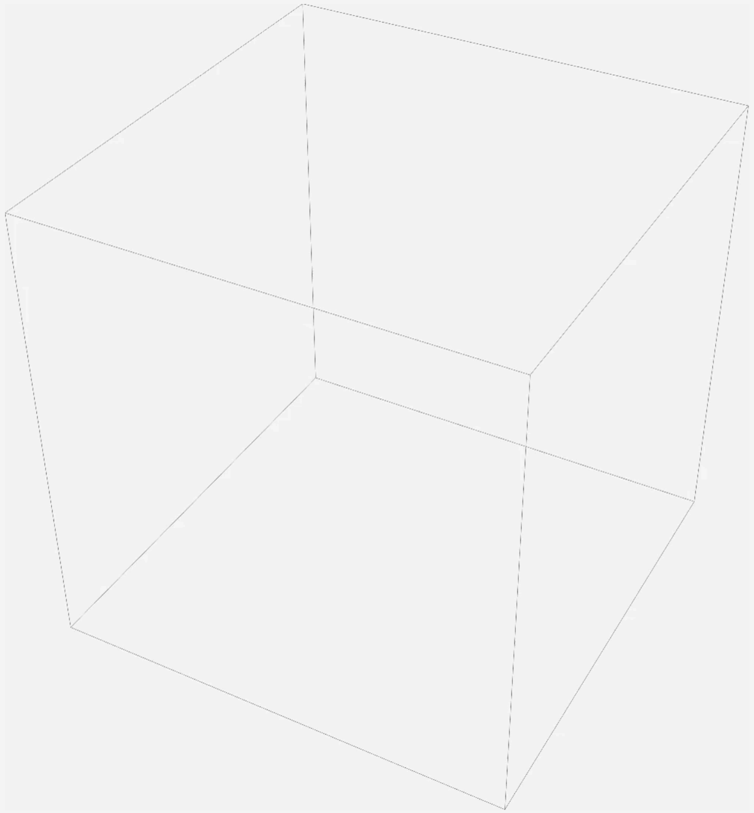


```
set(HDF5 ON CACHE BOOL "Set to ON to build with HDF5 [...] (default = OFF)")
```

(Also, you might need to include the MYHDF5_PATH (line 56) in CMakeLists.txt)

For more explanations you can consult Appendix A.3 of the CL manual.

Example

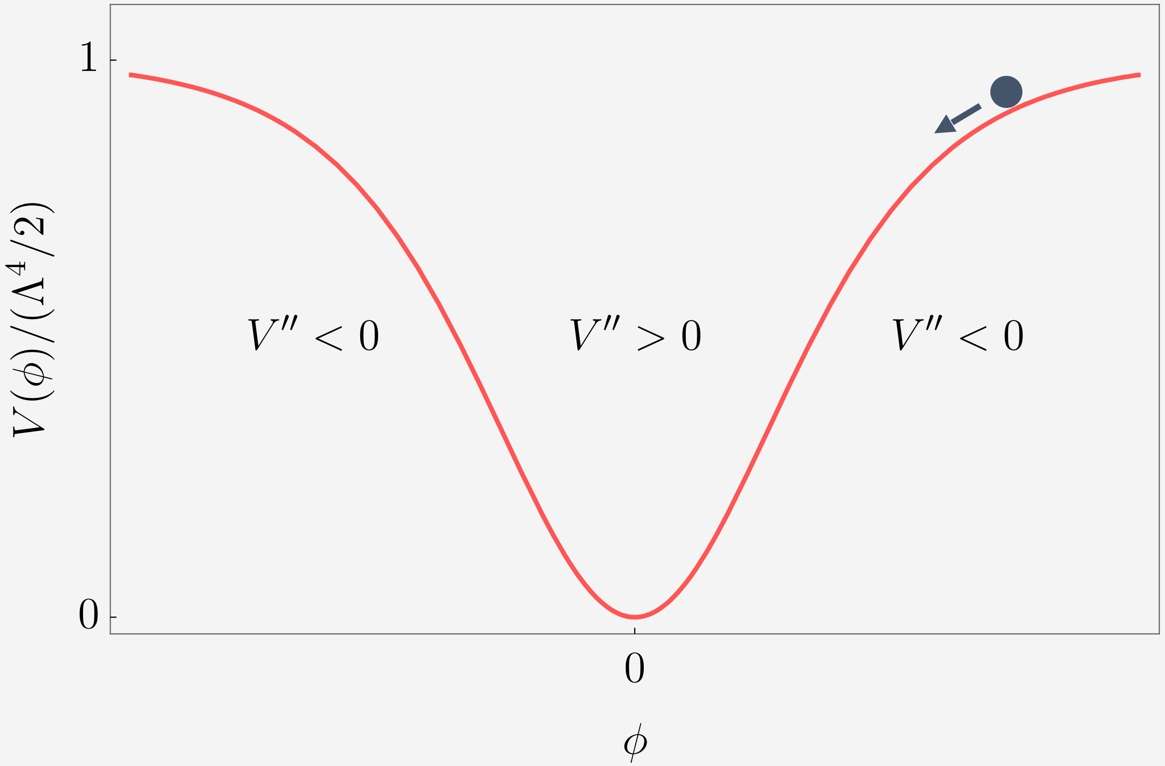


Example: Tachyonic Preheating

Potential we consider:

$$V(\phi) = \frac{1}{2}\Lambda^4 \tanh^2\left(\frac{\phi}{M}\right)$$

with $M = 0.01 m_{\rm pl}$



Example: Tachyonic Preheating

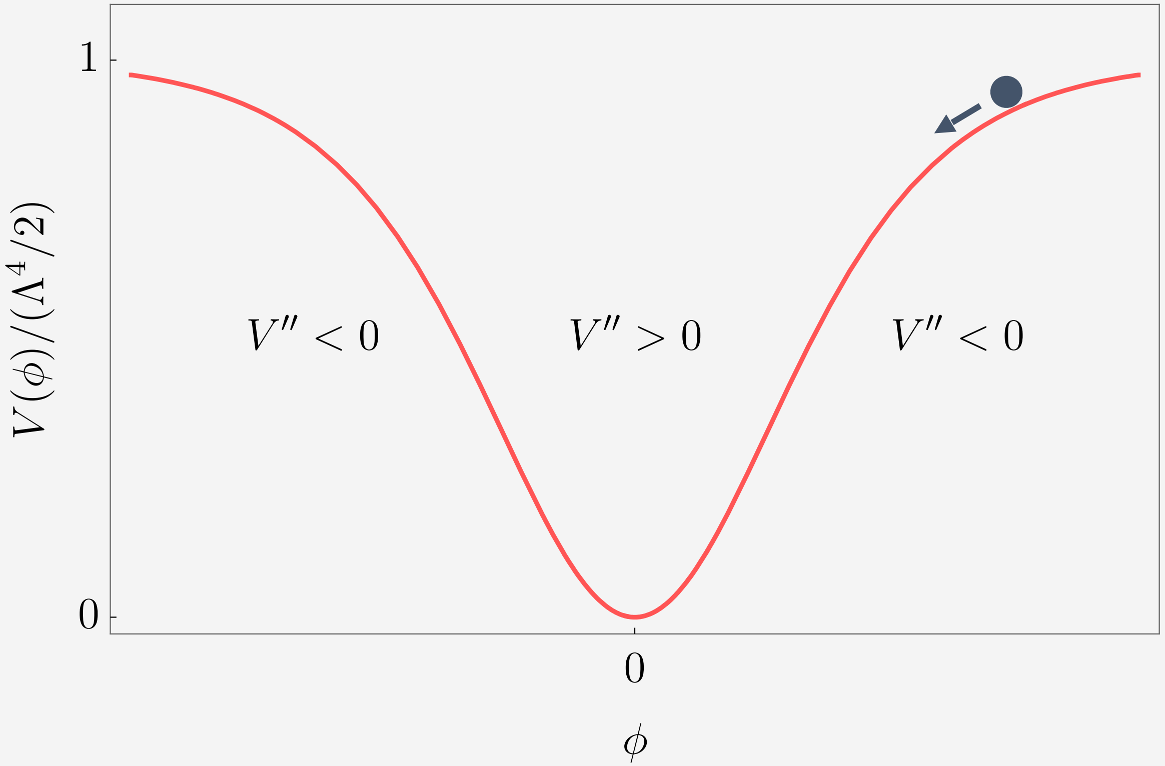
Potential we consider:

$$V(\phi) = \frac{1}{2}\Lambda^4 \tanh^2\left(\frac{\phi}{M}\right)$$

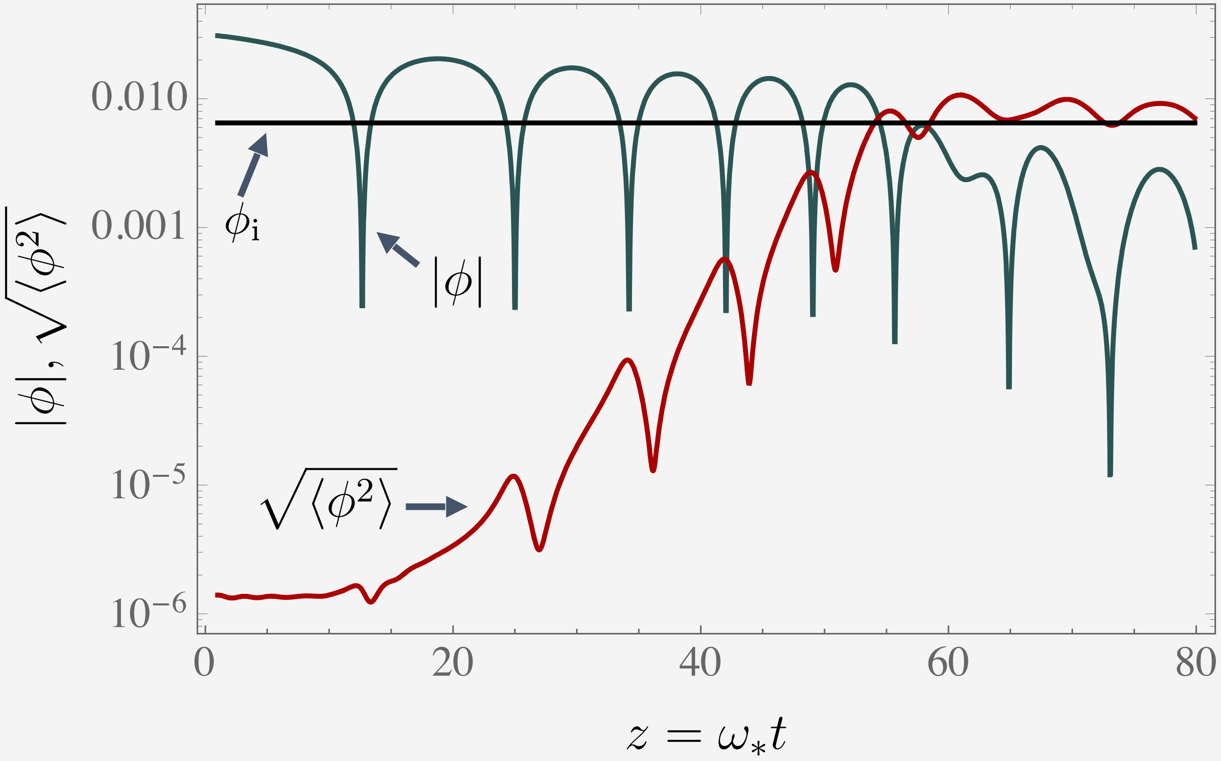
with $M = 0.01 m_{\rm pl}$

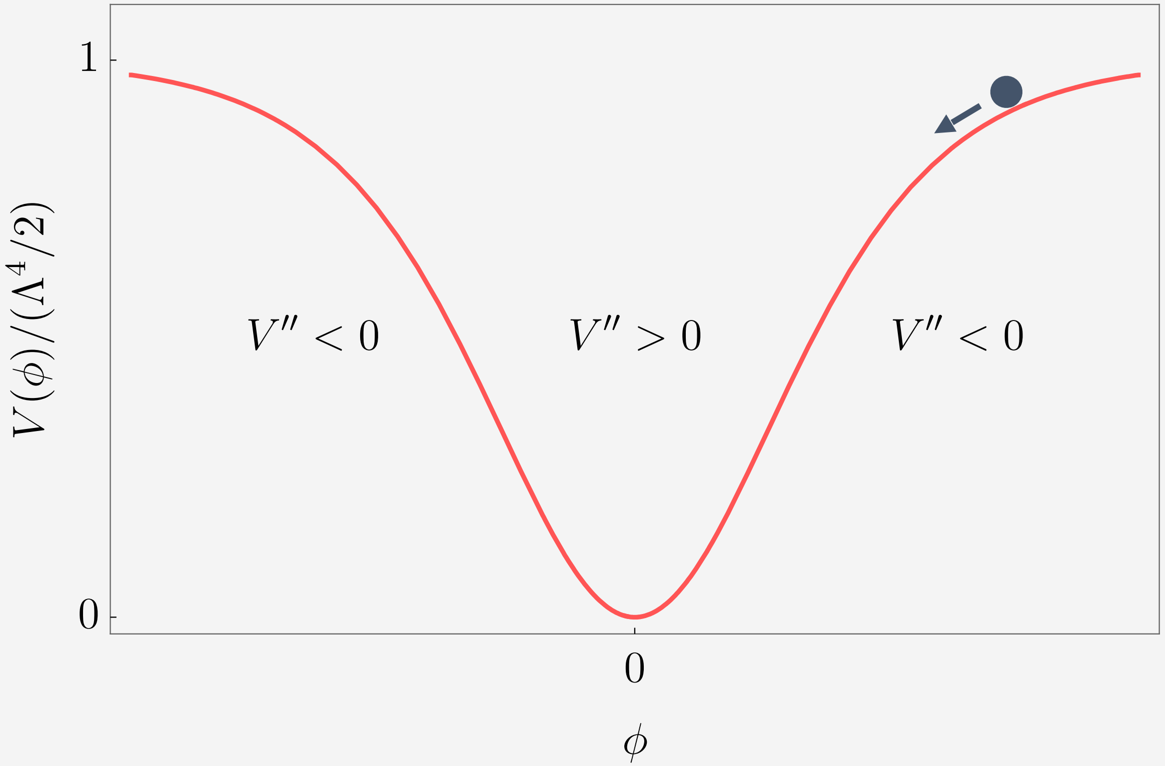
Inflection point:

$$\phi_{\rm i} = M \operatorname{arcsinh}\left(\sqrt{\frac{1}{2}}\right)$$



Example 1: Tachyonic Preheating





Example: Preparation

You can download from the CosmoLattice ‘25 School web page under Lecture 10 the following files:

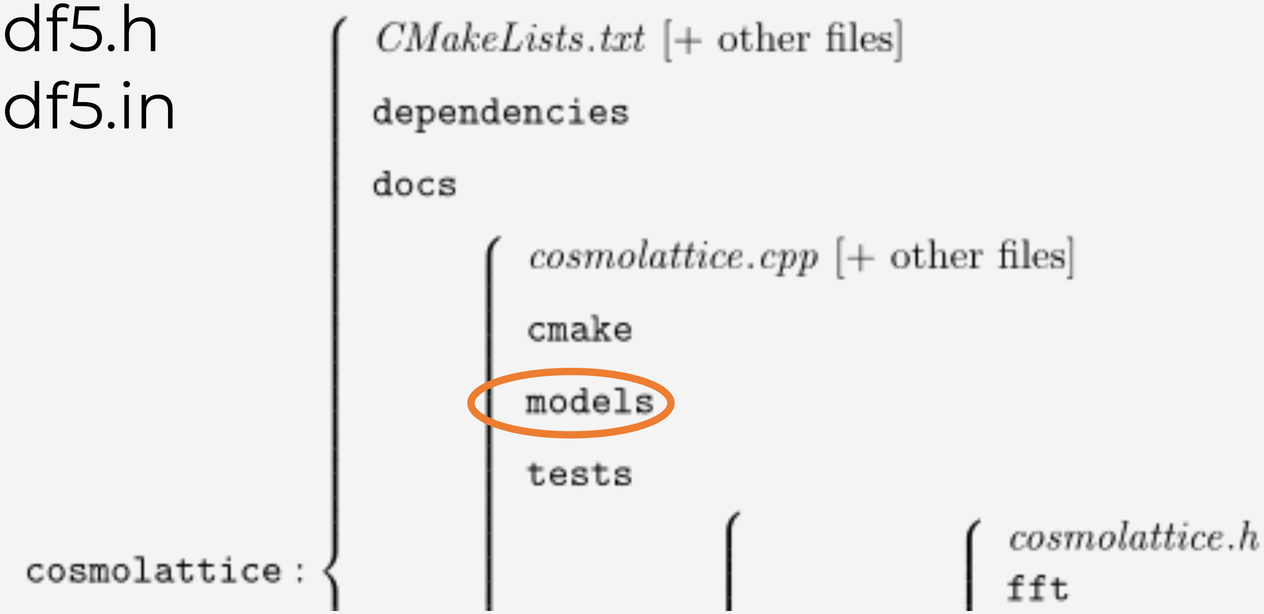
model file:	tanh2hdf5.h
input file:	tanh2hdf5.in
Mathematica notebook:	SnapshotPrinter.nb

Example 1: Preparation

You can download from the CosmoLattice ’25 School web page under Lecture 8 the following files:

model file:	tanh2hdf5.h
input file:	tanh2hdf5.in

Put the model file into the `models` folder and the input file into the `parameter-files` folder.



cosmolattice :

CMakeLists.txt [+ other files]

dependencies

docs

cosmolattice.cpp [+ other files]

cmake

models

tests

cosmolattice.h

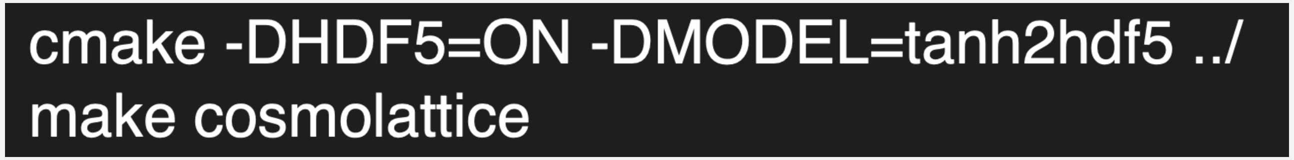
fft

Example: Compilation and Start Simulation

You can download from the CosmoLattice ’25 School web page under Lecture 8 the following files:

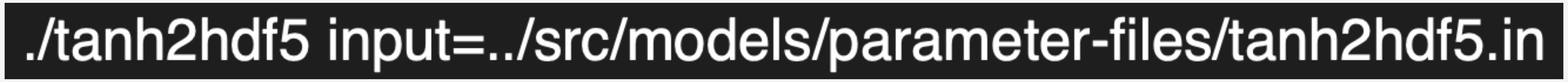
model file:	tanh2hdf5.h
input file:	tanh2hdf5.in

compile the model:



```
cmake -DHDF5=ON -DMODEL=tanh2hdf5 ../
make cosmolattice
```

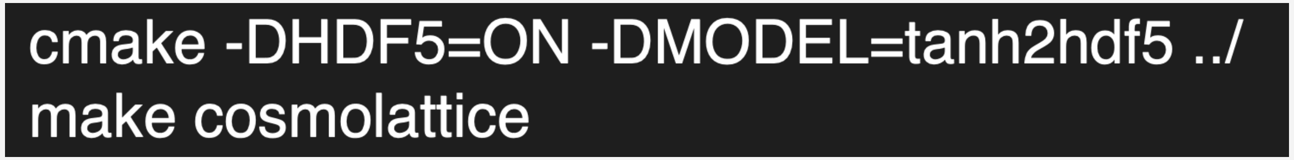
Start the simulation with:



```
./tanh2hdf5 input=../src/models/parameter-files/tanh2hdf5.in
```

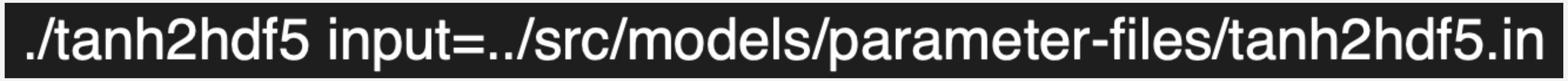
Example: Compilation and Run Simulation

Once the model and input files are placed correctly, we can open the terminal and compile the model by:



```
cmake -DHDF5=ON -DMODEL=tanh2hdf5 ../
make cosmolattice
```

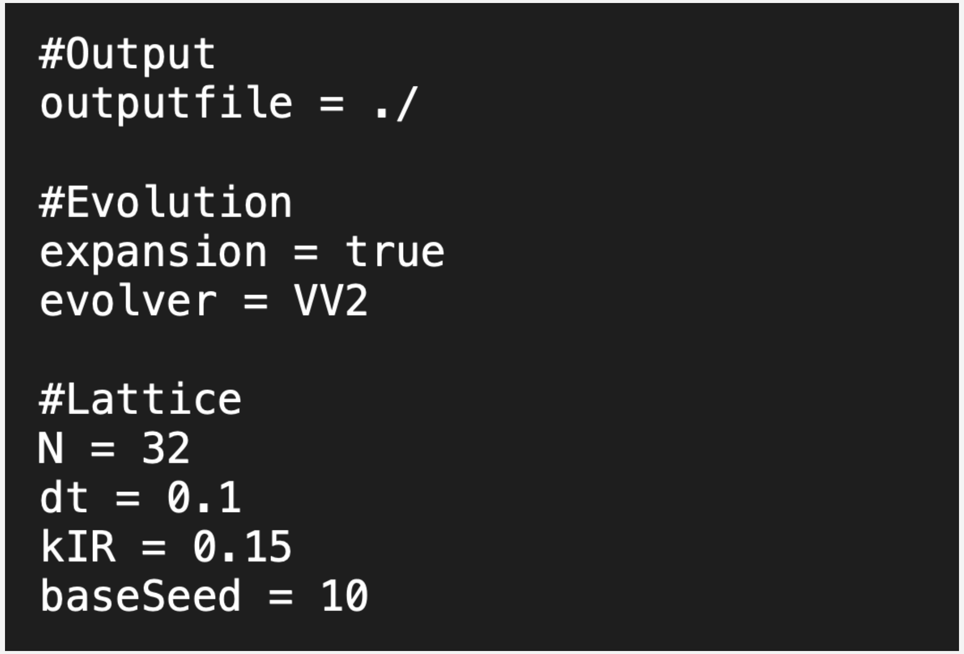
And then run the simulation with the following command:



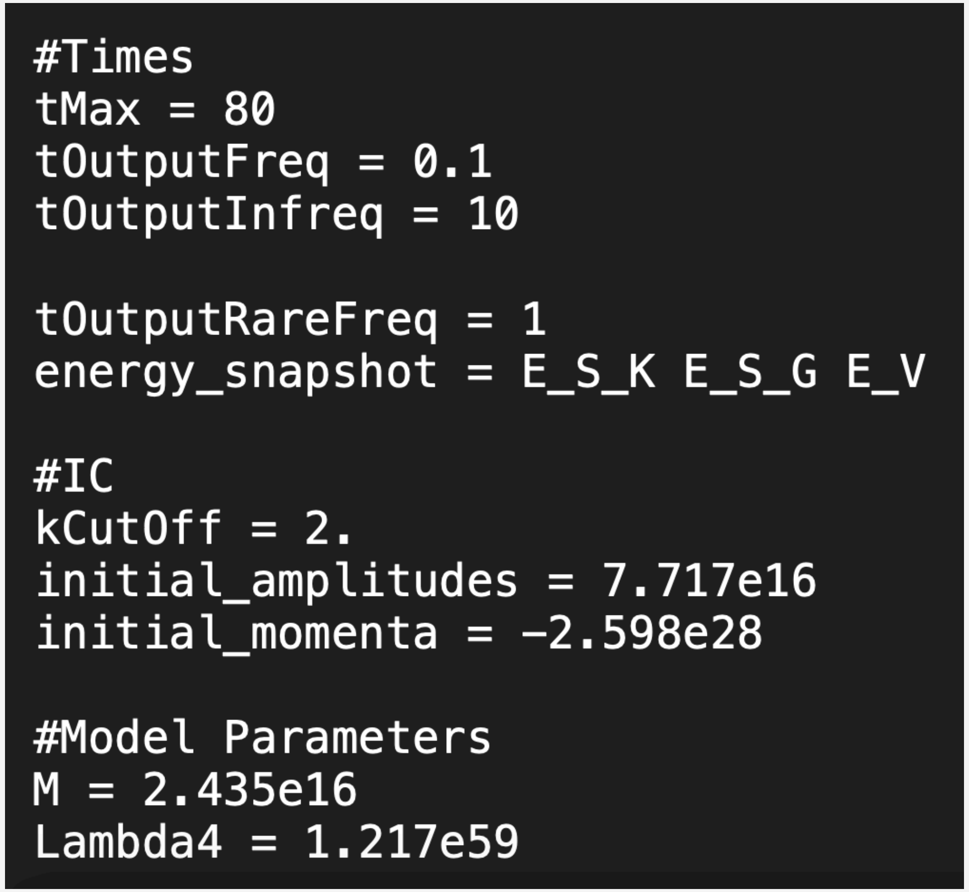
```
./tanh2hdf5 input=../src/models/parameter-files/tanh2hdf5.in
```

Example: Input file

What have we simulated? The input file tanh2hdf5.in contains the following simulation:



```
#Output
outputfile = ./

#Evolution
expansion = true
evolver = VV2

#Lattice
N = 32
dt = 0.1
kIR = 0.15
baseSeed = 10
```



```
#Times
tMax = 80
tOutputFreq = 0.1
tOutputInfreq = 10

tOutputRareFreq = 1
energy_snapshot = E_S_K E_S_G E_V

#IC
kCutOff = 2.
initial_amplitudes = 7.717e16
initial_momenta = -2.598e28

#Model Parameters
M = 2.435e16
Lambda4 = 1.217e59
```

Example: Output

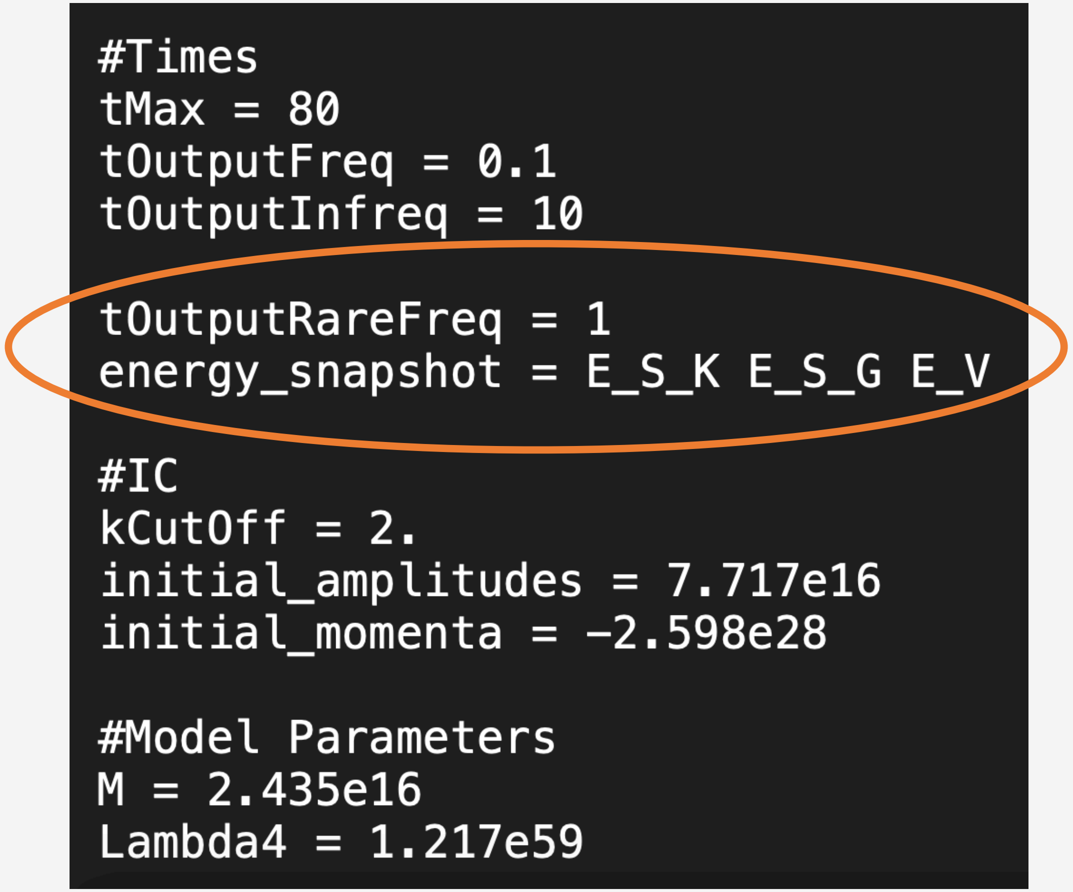
- By setting tOutputRareFreq = 1 at each timestep one snapshot is taken.

- The following energy snapshots are taken:

kinetic: $\tilde{K} = \frac{1}{2}(\tilde{\phi}')^2$

gradient: $\tilde{G} = \frac{1}{2}\sum_i (\tilde{\nabla}_i \tilde{\phi})^2$

potential: $\tilde{V}$



```
#Times
tMax = 80
tOutputFreq = 0.1
tOutputInfreq = 10

tOutputRareFreq = 1
energy_snapshot = E_S_K E_S_G E_V

#IC
kCutOff = 2.
initial_amplitudes = 7.717e16
initial_momenta = -2.598e28

#Model Parameters
M = 2.435e16
Lambda4 = 1.217e59
```

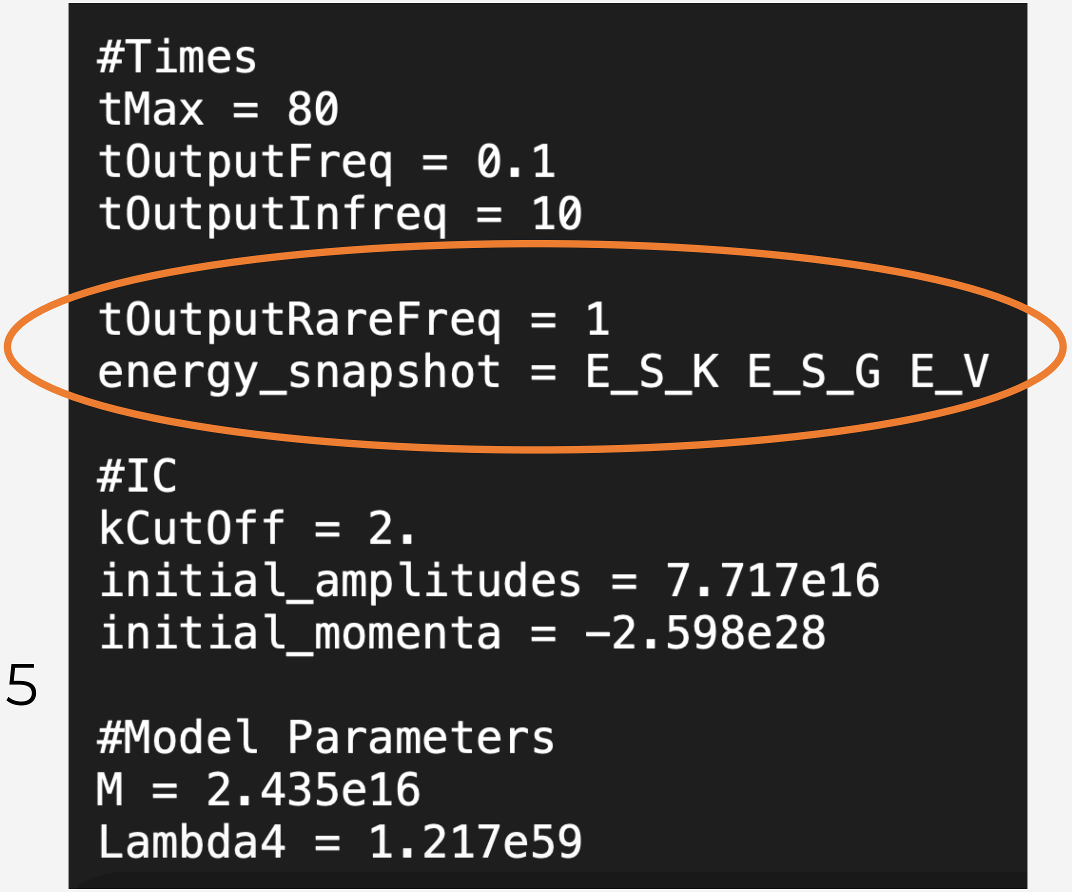
Example: Output

- By setting tOutputRareFreq = 1 at each time step one snapshot is taken.
- The following energy snapshots are taken:

  kinetic_energy_snapshot_scalar.h5

  gradient_energy_snapshot_scalar.h5

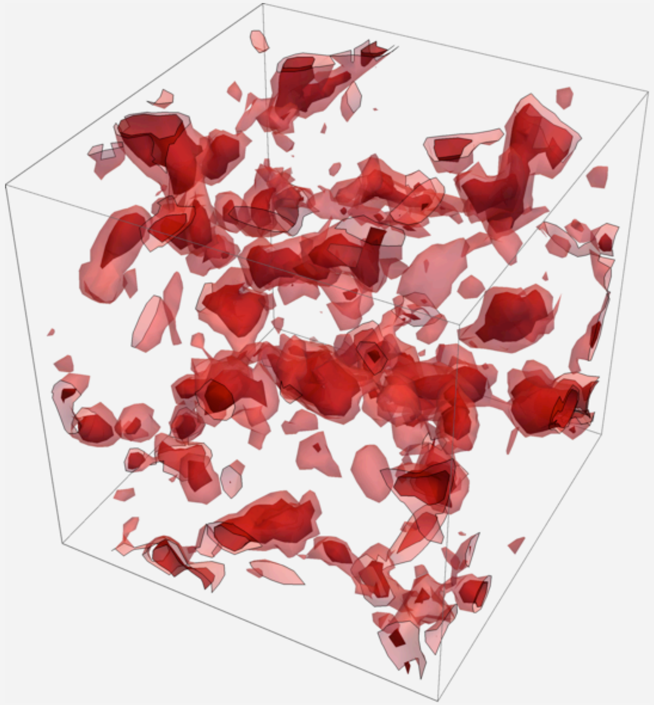
  potential_energy_snapshot.h5



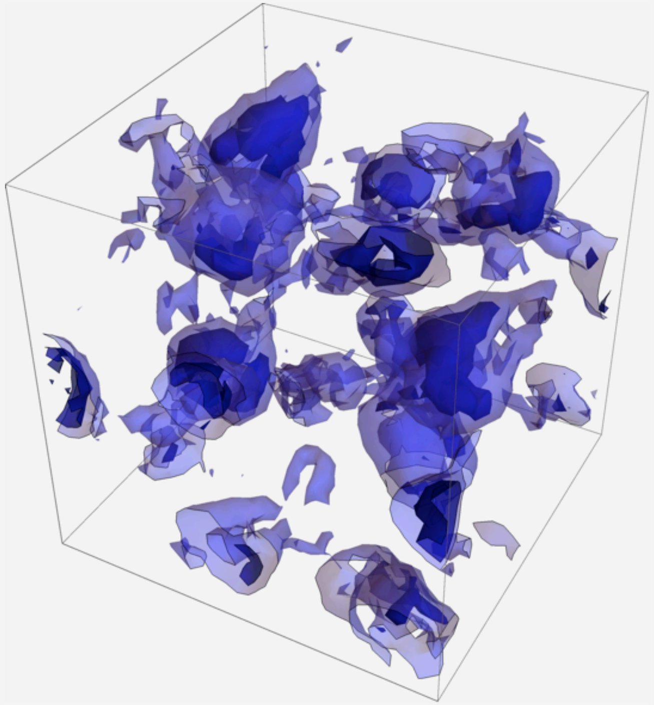
```
#Times
tMax = 80
tOutputFreq = 0.1
tOutputInfreq = 10

tOutputRareFreq = 1
energy_snapshot = E_S_K E_S_G E_V

#IC
kCutOff = 2.
initial_amplitudes = 7.717e16
initial_momenta = -2.598e28

#Model Parameters
M = 2.435e16
Lambda4 = 1.217e59
```

Example 1: Plot Energy Density Snapshots

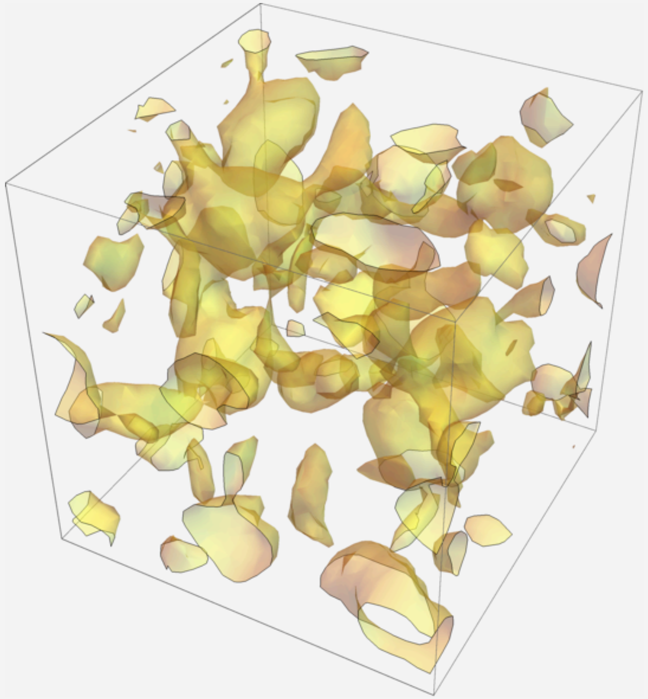
With the Mathematica notebook SnapshotPrinter.nb you can plot the different energy density snapshots:



$\varepsilon_k = \tilde{K}/\langle\tilde{E}_{\rm tot}\rangle$



$\varepsilon_g = \tilde{G}/\langle\tilde{E}_{\rm tot}\rangle$



$\varepsilon_p = \tilde{V}/\langle\tilde{E}_{\rm tot}\rangle$

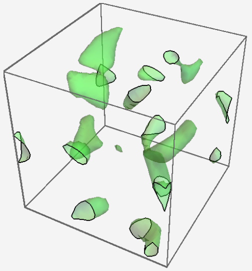
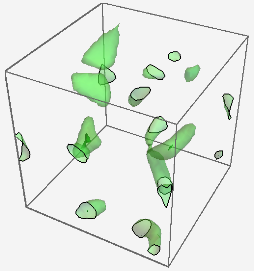
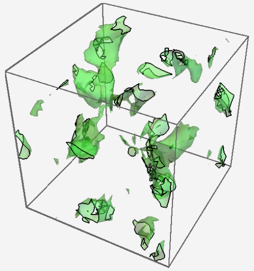
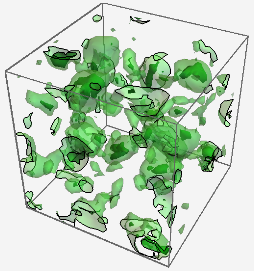
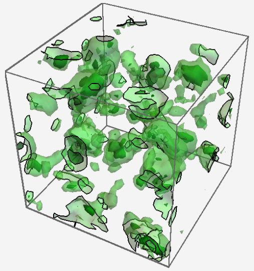
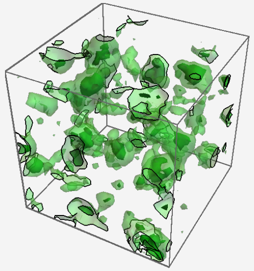
Example 1: Making a Movie

With the Mathematica notebook SnapshotPrinter.nb you can also make a movie of the evolution of the distribution of energy. For that purpose, we save all energy snapshots in a film reel (simply a list of plots):

```
                                 (tabKE0⟦i, 2⟧ + tabGE0⟦i, 2⟧ + tabVE0⟦i, 2⟧)
reel = Table[ListContourPlot3D[ ────────────────────────────────────────────── ,
                                            Etot[tabE0⟦i, 1⟧]
    Contours → {ContourA, ContourB}, ContourStyle → {ContourStyleA, ContourStyleB},
    PlotRange → All, Mesh -> None, Ticks → None], {i, 1, length}];
```

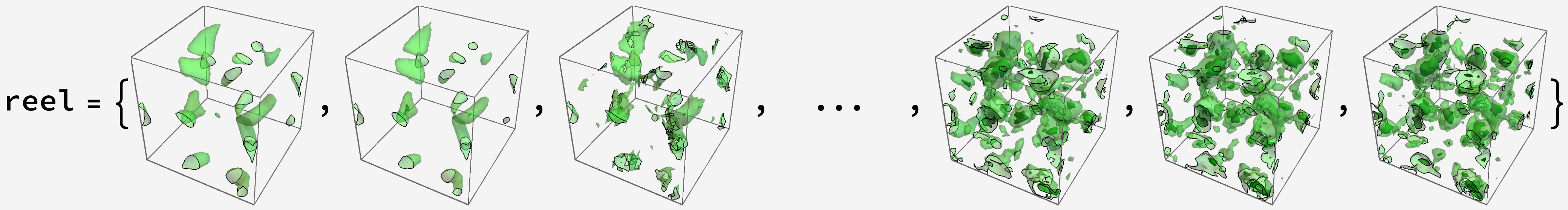
Example : Making a Movie

With the Mathematica notebook SnapshotPrinter.nb you can also make a movie of the evolution of the distribution of energy. For that purpose, we save all energy snapshots in a film reel (simply a list of plots):

reel = {  ,  ,  , . . . ,  ,  ,  }

Example: Making a Movie

With the Mathematica notebook SnapshotPrinter.nb you can also make a movie of the evolution of the distribution of energy. For that purpose, we save all energy snapshots in a film reel (simply a list of plots):



Which you can export as a movie:

```
Export["movie.avi", reel, "FrameRate" → 4, ImageResolution → 100]
```