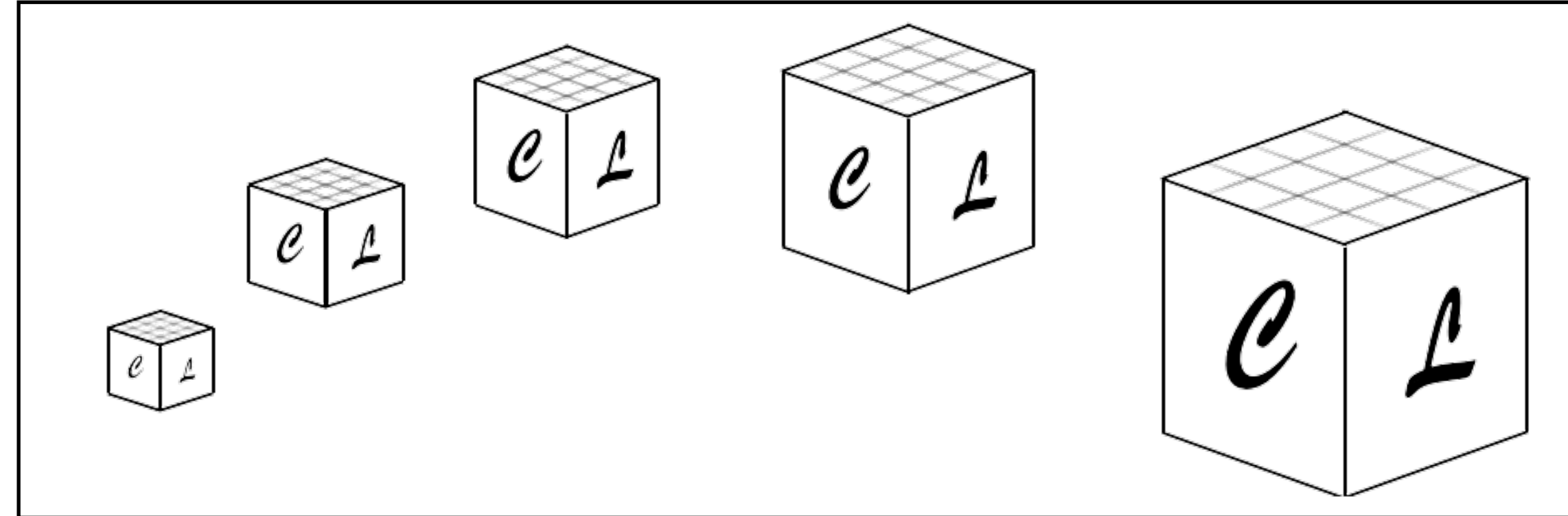


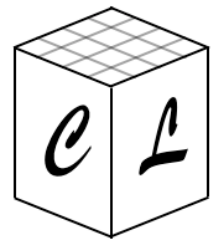


Cosmo $\mathcal{L}$ attice School:

Practice session 1: My first run with ϕ^4

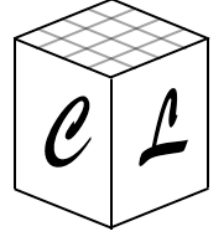
Nicolas Loayza, Kenneth Marschall and Franz Richard Sattler

Slides compiled by: Francisco Torrenti



Practice Session 1: Goals

- Get familiar with the basic folder structure of CL
- Compile and run simulations
- Analyse output with Mathematica, Jupyter Notebook, etc.



CosmoLattice installation

- Download the code: Type in the terminal:

```
git clone https://github.com/cosmolattice/cosmolattice
```

(or download it directly from the link)

- Prerequisites (OS X or linux):

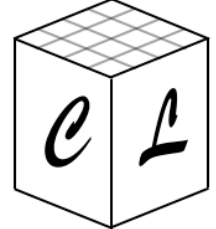
- CMake v3.0 (or above)
- g++ v5.0 (or above) or clang++ v3.4 (or above)
- fftw3 (note: it can be installed from our script)
- *For parallel use:* MPI.
- *Optional:* HDF5 and PFFT



**More info on Appendix A
of user manual!**

- Notebooks to produce figures: Download from INDICO's schedule

- Mathematica: <https://indico.ific.uv.es/event/7055/contributions/20588/attachments/10982/15683/L2b-lphi4.nb>
- Python: <https://indico.ific.uv.es/event/7055/contributions/20588/attachments/10982/15682/L2b-lphi4.ipynb>



CL: Basic commands

► Test run (in serial):

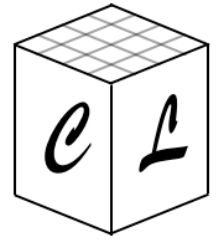
```
git clone https://github.com/cosmolattice/cosmolattice
cd cosmolattice
cd dependencies
bash fftw3.sh MyFFTW3
cd ..
mkdir build
cd build
cmake -DMODEL=lphi4 ../
make cosmolattice
./lphi4 input=../src/models/parameter-files/lphi4.in
```

Enter into main code folder
Enter into dependencies folder
Install FFTw3 from script
go back to main folder
Create a new directory
and go inside it
Selects model ϕ^4 for serial runs
Compiling
Executes serial run (input parameter file 'lphi4.in')

only if you want to install FFTw3 from the provided script

► To compile a second model (e.g. tanh2.h):

```
rm CMakeCache.txt
cmake - DMODEL=tanh2 ../
make cosmolattice
```



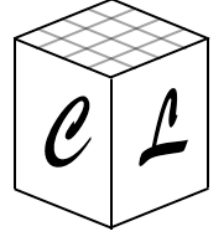
Example model: preheating in $\lambda\phi^4$

EXAMPLE MODEL:

$$S = \int d^4x a(t)^3 \mathcal{L}_m \quad -\mathcal{L}_m = \frac{1}{2} \partial^\mu \phi \partial_\mu \phi + \frac{1}{2} \partial^\mu \chi \partial_\mu \chi + V(\phi, \chi)$$

$$V(\phi, \chi) \equiv \sum_{m=0}^{N_p-1} V^{(m)}(\phi, \chi) = \frac{\lambda}{4} \phi^4 + \frac{1}{2} g^2 \phi^2 \chi^2$$

- Number scalar singlets: $N_s = 2$
- Number potential terms: $N_p = 2$



Example model: preheating in $\lambda\phi^4$

► **Program variables:** $d\tilde{\eta} \equiv a^{-\alpha} \omega_* dt$ $d\tilde{x}^i \equiv \omega_* dx^i$ $\tilde{\phi} = \frac{\phi}{f_*}$

$$\begin{aligned} f_* &= \phi_* \\ \omega_* &= \sqrt{\lambda} \phi_* \\ \alpha &= 1 \end{aligned}$$



$$\begin{aligned} d\tilde{\eta} &\equiv \sqrt{\lambda} \phi_* a^{-1} dt & d\tilde{x}^i &\equiv \sqrt{\lambda} \phi_* dx^i \\ \tilde{\phi} &= \frac{\phi}{\phi_*} & \tilde{\chi} &= \frac{\chi}{\phi_*} \end{aligned}$$

Program variables for $V(\phi) = (\lambda/4)\phi^4$

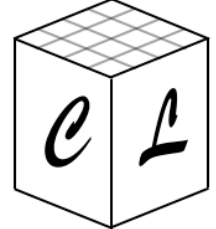
► **Program potential:**

$$\tilde{V}(\tilde{\phi}, \tilde{\chi}) \equiv \frac{1}{f_*^2 \omega_*^2} V(f_* \tilde{\phi}, f_* \tilde{\chi}) = \underbrace{\frac{1}{4} |\tilde{\phi}|^4}_{\tilde{V}^{(0)}} + \underbrace{\frac{1}{2} q \tilde{\phi}^2 \tilde{\chi}^2}_{\tilde{V}^{(1)}}$$

$$q \equiv \frac{g^2}{\lambda}$$

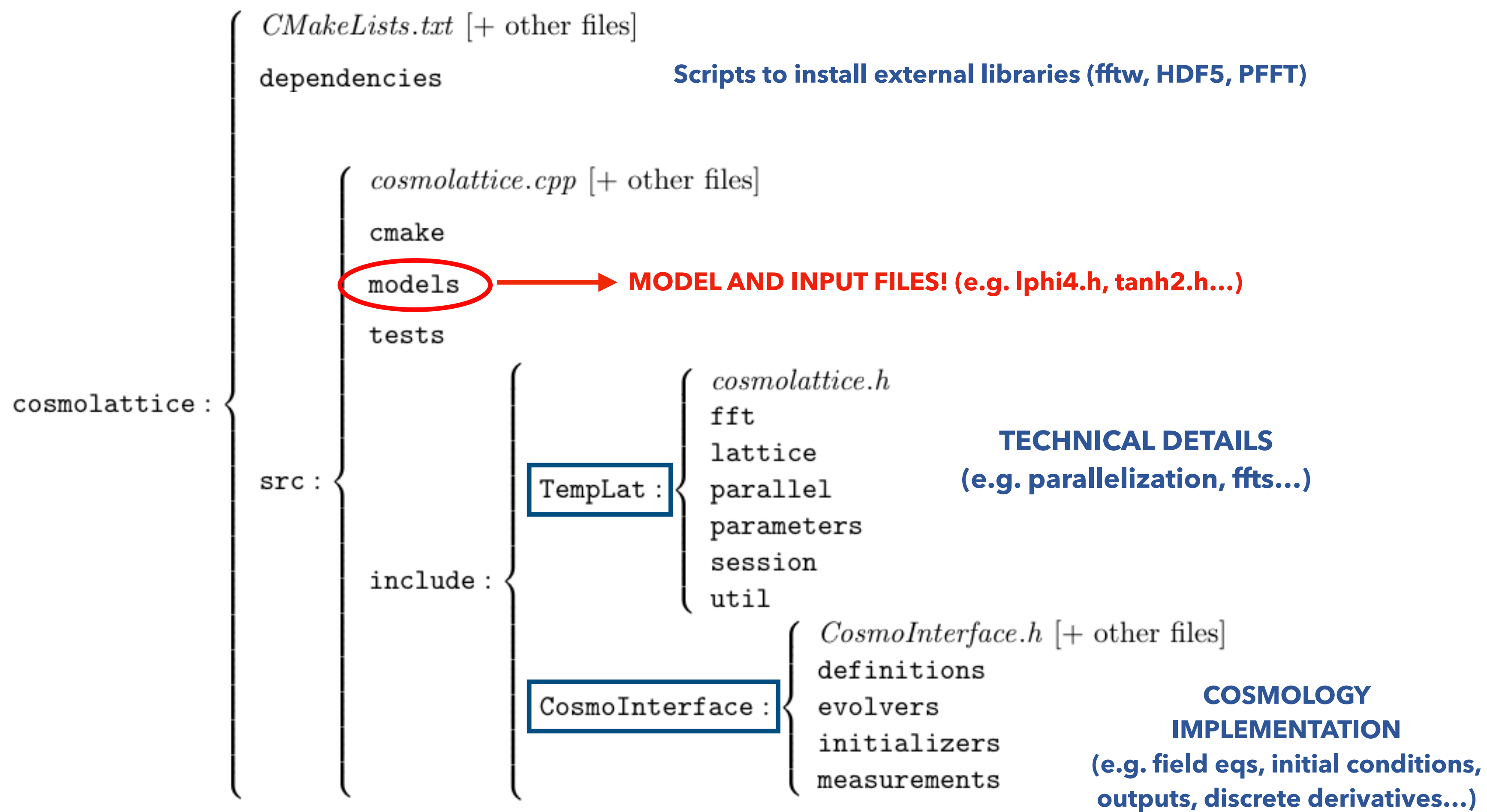
**Resonance
parameter**

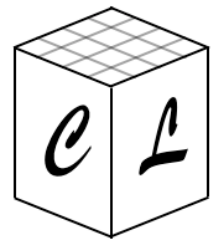
**MODEL ALREADY
IMPLEMENTED IN THE CODE!**



CL folder tree structure

► Basic folder tree structure:

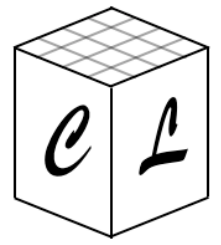




Model implementation: files

Implementation of a model only requires handling two files:

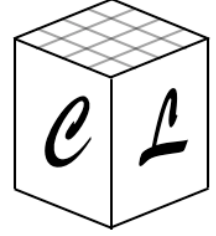
- The model file
- The parameter (or “input”) file



Model implementation: files

Implementation of a model only requires handling two files:

- The model file
- The parameter (or "input") file



Model file: lphi4.h

$$\tilde{V}(\tilde{\phi}, \tilde{\chi}) = \frac{1}{4}\tilde{\phi}^4 + \frac{1}{2}q\tilde{\phi}^2\tilde{\chi}^2$$

lphi4.h

```
1  #ifndef LPHI4_H
2  #define LPHI4_H
3
4  #include "CosmoInterface/cosmointerface.h"
5
6  namespace TempLat
7  {
8      // Model name and number of fields
9      struct ModelPars : public TempLat::DefaultModelPars {
10         static constexpr size_t NScalars = 2;
11         static constexpr size_t NPotTerms = 2;
12     };
13
14     #define MODELNAME lphi4 // model name
15
16     template<class R>
17     using Model = MakeModel(R, ModelPars);
18
19     class MODELNAME : public Model<MODELNAME>
20     {
```

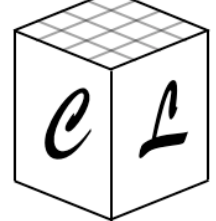
Number of scalar fields

$\tilde{\phi}$ field 0
 $\tilde{\chi}$ field 1

Number of terms in potential

$$\tilde{V}^{(0)} = \frac{1}{4}\tilde{\phi}^4 \quad \text{term 0}$$
$$\tilde{V}^{(1)} = \frac{1}{2}q\tilde{\phi}^2\tilde{\chi}^2 \quad \text{term 1}$$

Model name: it has to match the name of the file!



Model file: lphi4.h

```
22 private:
23     double g, lambda, q;
24
25 public:
26
27     MODELNAME(ParameterParser& parser, RunParameters<double>& runPar,
28     Model<MODELNAME>(parser, runPar.getLatParams(), toolBox, runPar.dt
29 {
30
31     // Independent parameters of the model
32     lambda = parser.get<double>("lambda");
33     q = parser.get<double>("q");
34
35     g = sqrt(q*lambda);
36
37     // Initial homogeneous components of the fields
38     // (read from parameters file, or specified here if not)
39     fldS0 = parser.get<double, 2>("initial_amplitudes");
40     piS0 = parser.get<double, 2>("initial_momenta", {0, 0});
41
42     // Rescaling for program variables
43     alpha = 1;
44     fStar = fldS0[0];
45     omegaStar = sqrt(lambda)*fStar;
46
47     setInitialPotentialAndMassesFromPotential();
48
49 }
```

Model parameters g, λ, q

Parameters given in input file λ, q

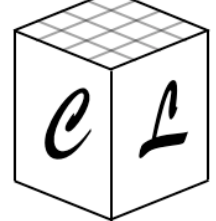
Derived parameters $g = \sqrt{q\lambda}$

**Vectors of initial
homogeneous components**
 (ϕ_*, χ_*) $(\dot{\phi}_*, \dot{\chi}_*)$

Default values

Program variables

$\alpha = 1$ $f_* = \phi_*$ $\omega_* = \sqrt{\lambda}\phi_*$



Model file: lphi4.h

$$\tilde{V}(\tilde{\phi}, \tilde{\chi}) = \frac{1}{4}\tilde{\phi}^4 + \frac{1}{2}q\tilde{\phi}^2\tilde{\chi}^2$$

```
52 // Program potential
53 auto potentialTerms(Tag<0>) // term 0
54 {
55     return 0.25 * pow<4>(fldS(0_c));
56 }
57 auto potentialTerms(Tag<1>) // term 1
58 {
59     return 0.5 * q * pow<2>(fldS(0_c) * fldS(1_c));
60 }
```

Potential terms:

$$\tilde{V}^{(0)} = \frac{1}{4}\tilde{\phi}^4$$

$$\tilde{V}^{(1)} = \frac{1}{2}q\tilde{\phi}^2\tilde{\chi}^2$$

```
62 // Derivatives of the program potential with respect fields
63 auto potDeriv(Tag<0>) // dV/dphi
64 {
65     return pow<3>(fldS(0_c)) + q * fldS(0_c) * pow<2>(fldS(1_c)) ;
66 }
67 auto potDeriv(Tag<1>) // dV/dchi
68 {
69     return q * fldS(1_c) * pow<2>(fldS(0_c));
70 }
```

First derivatives:

$$\tilde{V}_{,\tilde{\phi}} = \tilde{\phi}^3 + q\tilde{\chi}^2\tilde{\phi}$$

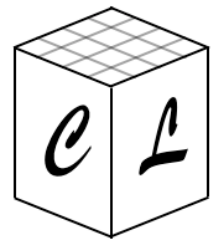
$$\tilde{V}_{,\tilde{\chi}} = q\tilde{\phi}^2\tilde{\chi}$$

```
72 // Second derivatives of the program potential with respect fields
73 auto potDeriv2(Tag<0>) // d2V/dphi2
74 {
75     return 3 * pow<2>(fldS(0_c)) + q * pow<2>(fldS(1_c)) ;
76 }
77 auto potDeriv2(Tag<1>) // d2V/dchi2
78 {
79     return q * pow<2>(fldS(0_c)) ;
80 }
```

Second derivatives:

$$\tilde{V}_{,\tilde{\phi}\tilde{\phi}} = 3\tilde{\phi}^2 + q\tilde{\chi}^2$$

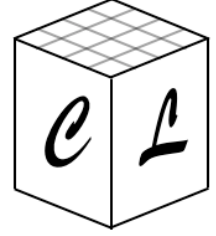
$$\tilde{V}_{,\tilde{\chi}\tilde{\chi}} = q\tilde{\phi}^2$$



Model implementation: files

Implementation of a model only requires handling two files:

- The model file
- The parameter (or "input") file



Input file: lphi4.in

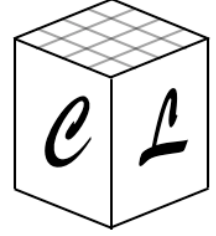
lphi4.in

1	#Output				
2	outputfile = ./	→	Output folder: Path where output files are created		(./ : execution folder)
3					
4	#Evolution				
5	expansion = true	→	Expansion: true or false		
6	evolver = VV2	→	Evolvers: VV2, VV4, VV6, VV8, VV10, LF2		
7					
8	#Lattice				
9	N = 32	→	Number of points per dimension: N		
10	dt = 0.01	→	Time step: $\delta\tilde{\eta}$		
11	kIR = 0.75	→	Infrared cutoff of the lattice: $\tilde{k}_{\text{IR}} = k_{\text{IR}}/\omega_*$		
12					
13	#Times				
14	tOutputFreq = 0.1	→	Frequency of frequent output (for volume averages)		
15	tOutputInfreq = 1	→	Frequency of infrequent output (for spectra)		
16	tMax = 300	→	Final time of simulation		
17					

in program units!

↓

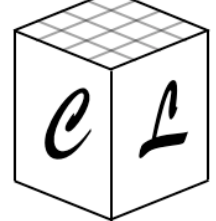
RULE OF THUMB: If ω_* is set to the oscillation frequency, these numbers should be close to 1



Input file: lphi4.in

lphi4.in

18	#Spectra options		Type and version of power spectrum
19	PS_type = 1	→	(See Technical Note I)
20	PS_version = 1		
21			
22	#GWs		Gravitational waves
23	GWprojectorType = 1	→	(See Lecture 6)
24	withGWs=false		
25			
26	#IC		Cutoff for initial fluctuations
27	kCutOff = 4	→	(modes with $k > k_{\text{CutOff}}$ are not populated at the initial time)
28	initial_amplitudes = 5.6964e18 0		
29	initial_momenta = -4.86735e30 0	→	Initial homogeneous values: for field amplitudes and time-derivatives (In GeV!!)
30			
31	#Model Parameters		
32	lambda = 9e-14	→	Model parameters: defined in model file
33	q = 100		



Output produced by CL (scalar model)

➤ [model_name].infos: Information about the run.

➤ average_scale_factor.txt: $\tilde{\eta}, a, a', a'/a$

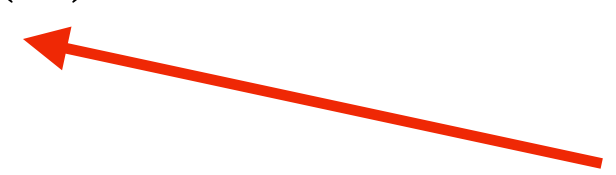
➤ average_scalar_[n].txt: $\tilde{\eta}, \langle \tilde{\phi}_n \rangle, \langle \tilde{\phi}'_n \rangle, \langle \tilde{\phi}_n^2 \rangle, \langle \tilde{\phi}'_n{}^2 \rangle, rms(\tilde{\phi}_n), rms(\tilde{\phi}'_n)$

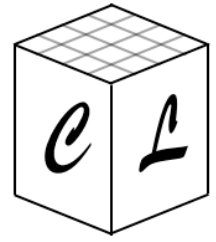
➤ average_energy_conservation.txt: $\tilde{\eta}, \frac{\langle LHS - RHS \rangle}{\langle LHS + RHS \rangle}, \langle LHS \rangle, \langle RHS \rangle$


$$\text{LHS} \quad a'^2 = \frac{a^{2\alpha+2}}{3} \left(\frac{f_*}{m_p} \right)^2 \tilde{\rho} \quad \text{RHS}$$

➤ average_energies.txt: $\tilde{\eta}, \tilde{E}_K^{(0)}, \tilde{E}_G^{(0)}, \dots, \tilde{E}_K^{(N_s-1)}, \tilde{E}_G^{(N_s-1)}, \tilde{E}_V^{(0)}, \dots, \tilde{E}_V^{(N_p-1)}, \langle \tilde{\rho} \rangle$

➤ spectra_scalar_[n].txt: $\tilde{k}, \tilde{\Delta}_{\tilde{\phi}}(\tilde{k}), \tilde{\Delta}_{\tilde{\phi}'}(\tilde{k}), \Delta n_{bin}$


$$\tilde{\Delta}_{\tilde{\phi}} \equiv \frac{\Delta_{\phi}}{f_*^2} \quad \tilde{\Delta}_{\tilde{\phi}'} \equiv \frac{\Delta_{\phi'}}{f_*^2 \omega_*^2}$$

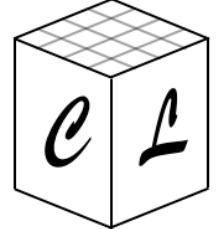


Exercise 1

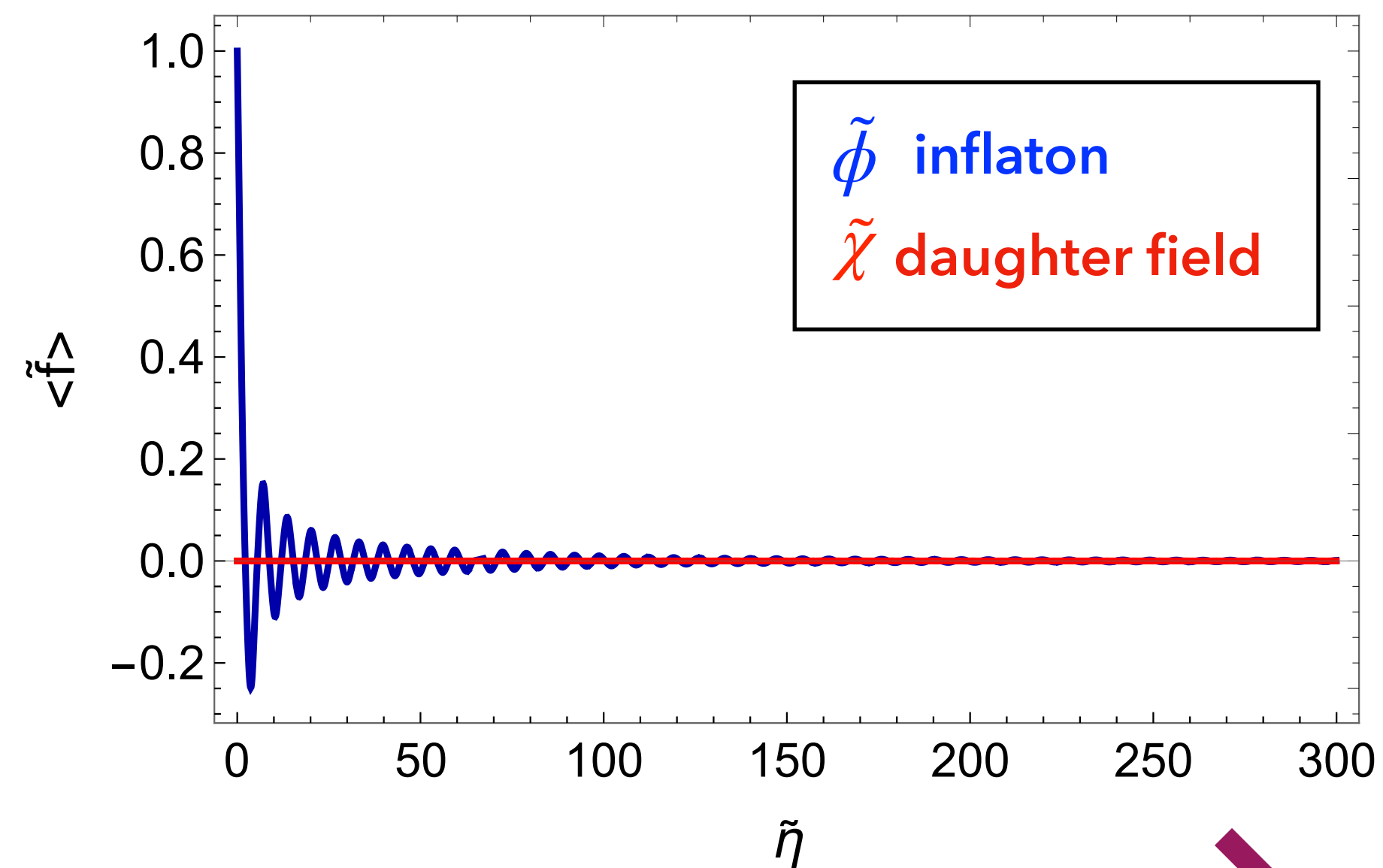
1. Run the **lphi4** model with the default parameters (the ones provided in the lphi4.in input file). Plot figures for the volume-averaged field amplitudes and variances, energy components, and field spectra. You can use the provided mathematica or python notebooks.

```
cd cosmolattice
cd build
cmake -DMODEL=lphi4 ../
make cosmolattice
../lphi4 input=../src/models/parameter-files/lphi4.in
```

Lattice parameters:	Model parameters:
$N = 32$	$q \equiv g^2/\lambda = 100$
$\delta\tilde{\eta} = 0.01$	$\lambda = 9 \cdot 10^{-14}$
$\tilde{k}_{\text{IR}} = 0.75$	



Output: field amplitudes



Lattice
parameters:

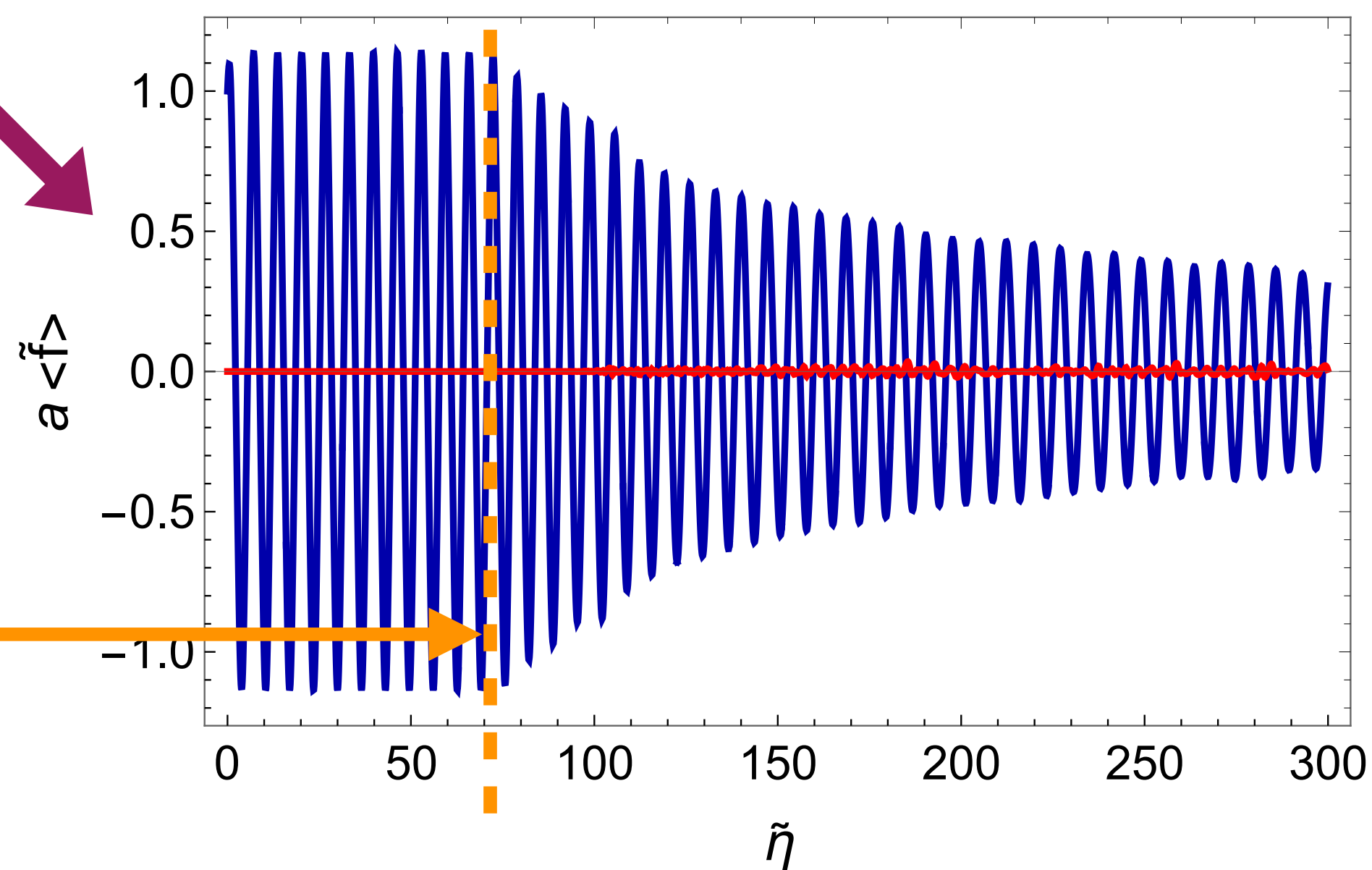
$$N = 32$$
$$\delta\tilde{\eta} = 0.01$$
$$\tilde{k}_{\text{IR}} = 0.75$$

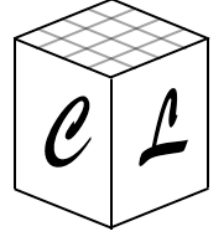
Model
parameters:

$$q \equiv g^2/\lambda = 100$$
$$\lambda = 9 \cdot 10^{-14}$$

conformal
transformation

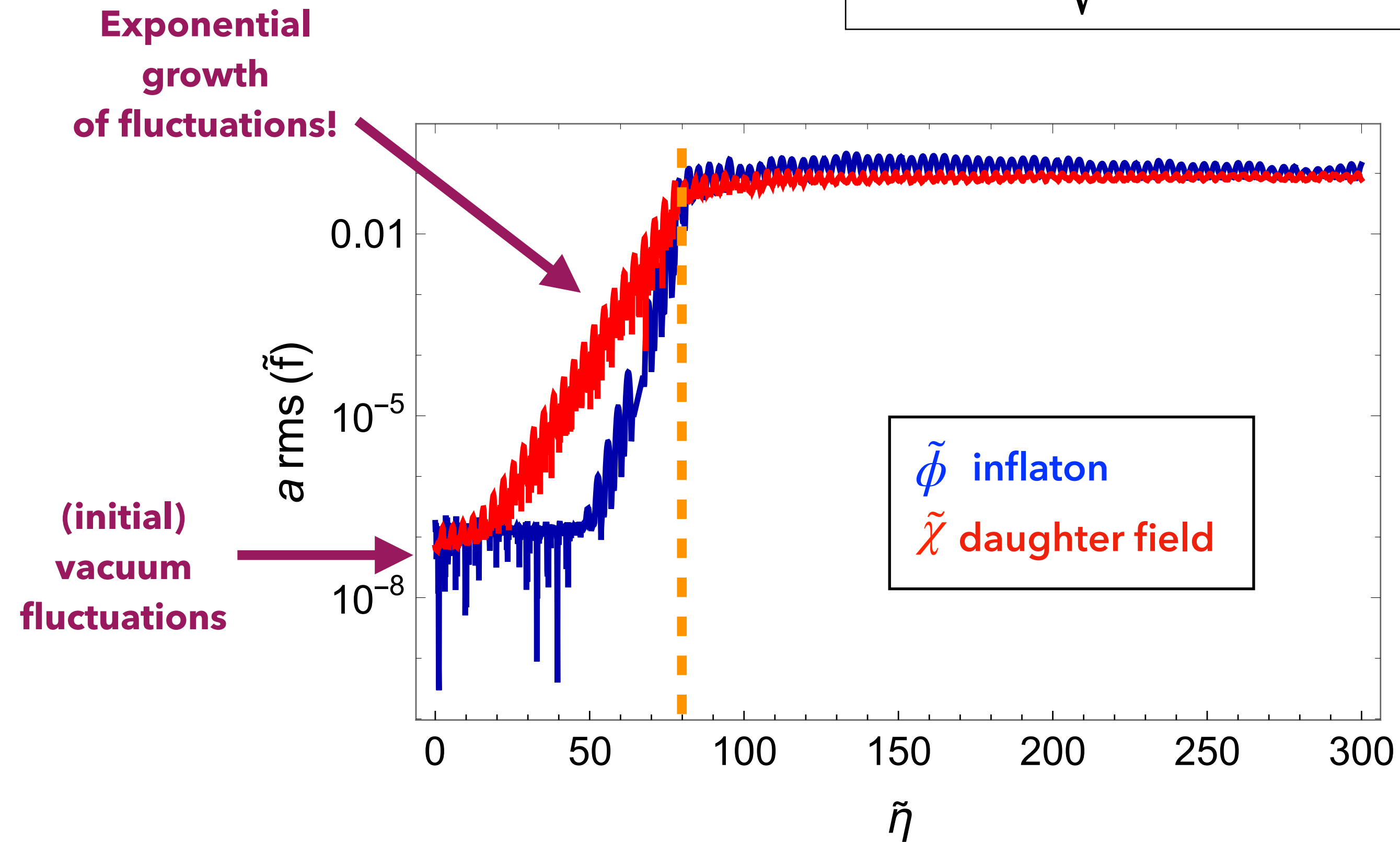
start of non-linear
regime



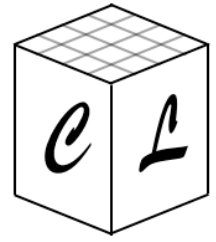


Output: variances

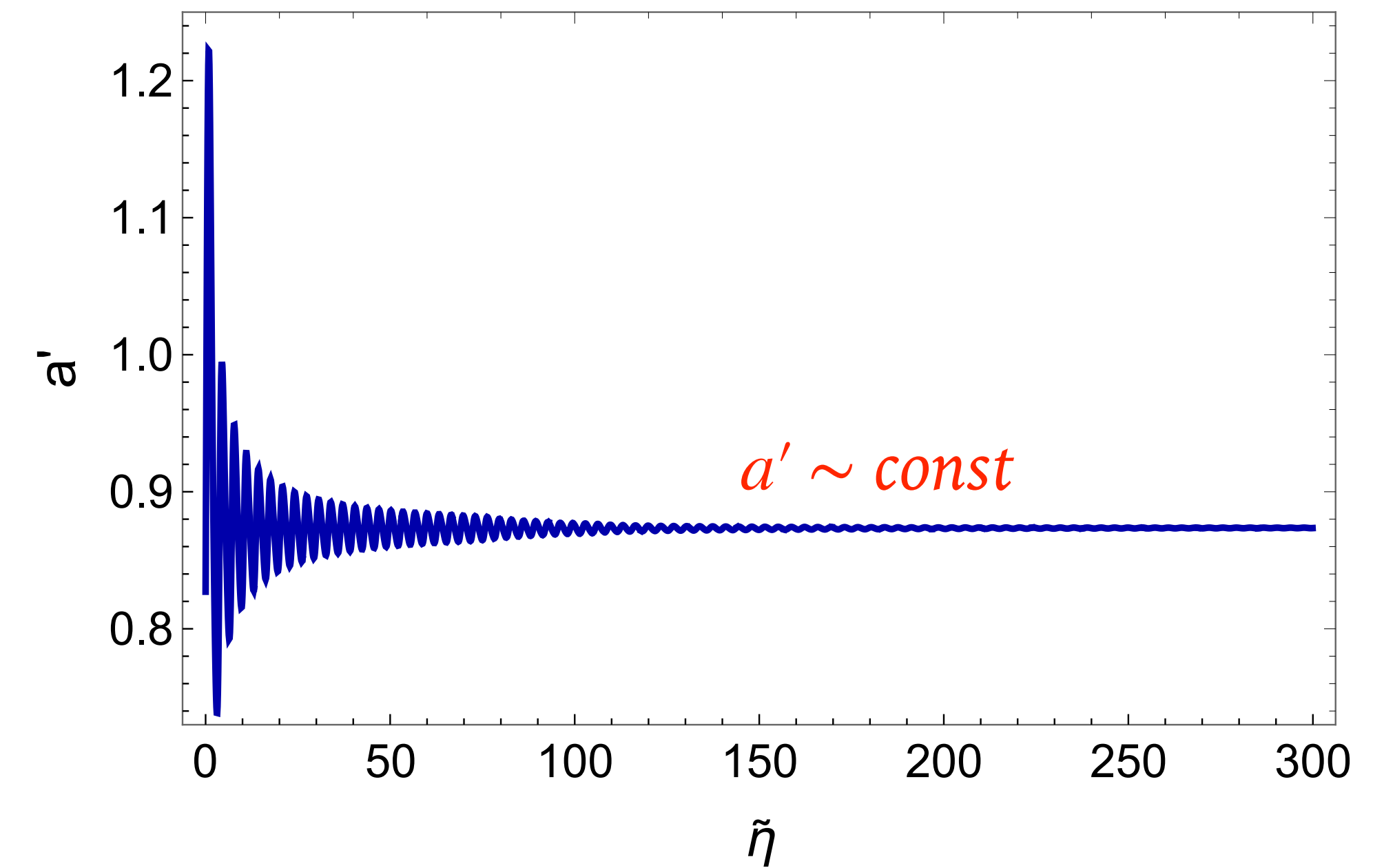
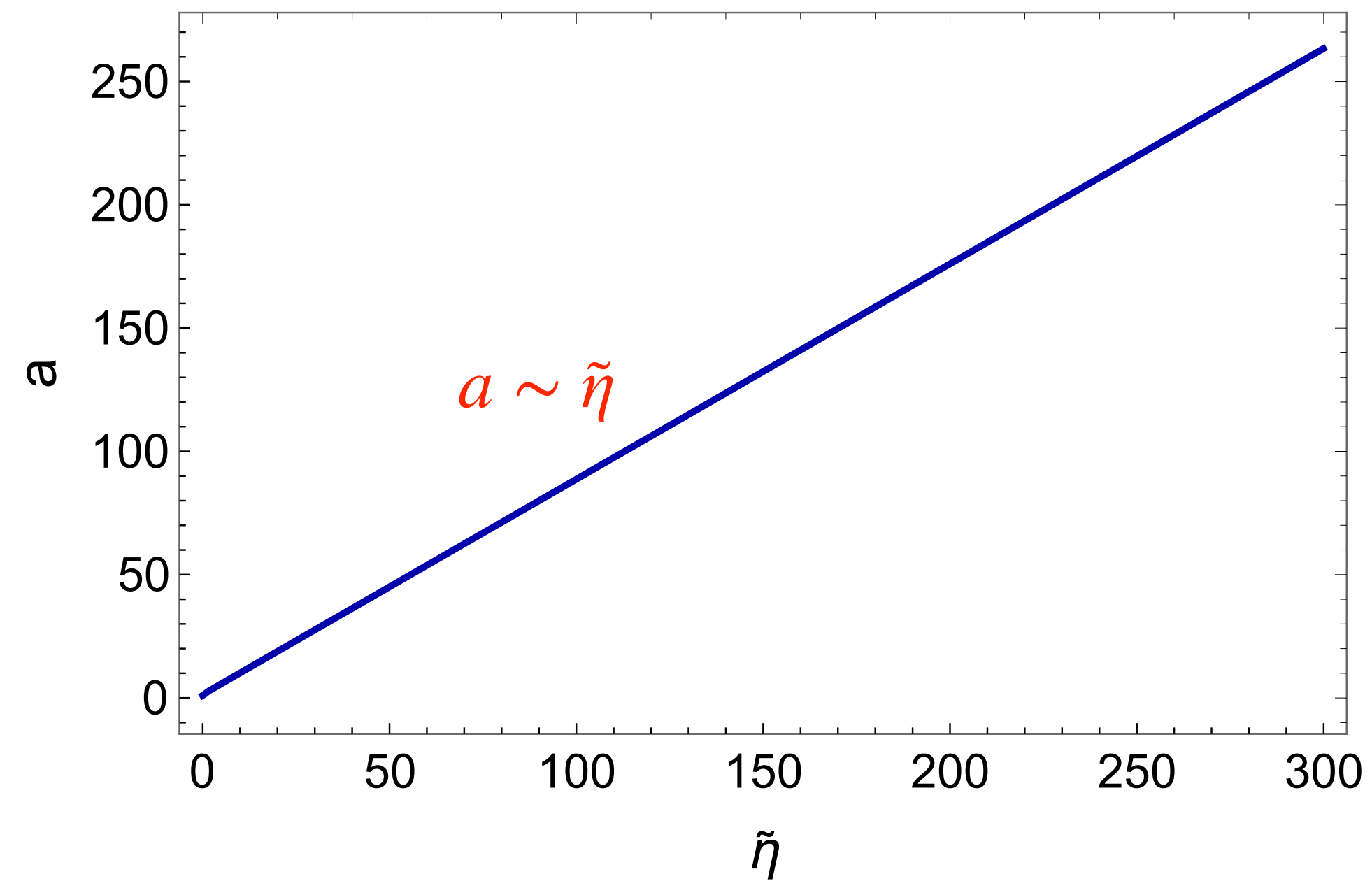
$$rms(\tilde{f}) \equiv \sqrt{\langle \tilde{f}^2 \rangle - \langle \tilde{f} \rangle^2} \quad \tilde{f} = \tilde{\phi}, \tilde{\chi}$$



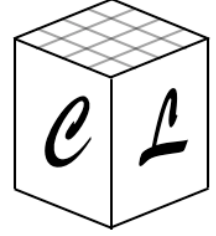
The excitation of the **daughter field** during the linear regime is stronger than for the inflaton, and dominates the energy budget



Output: scale factor

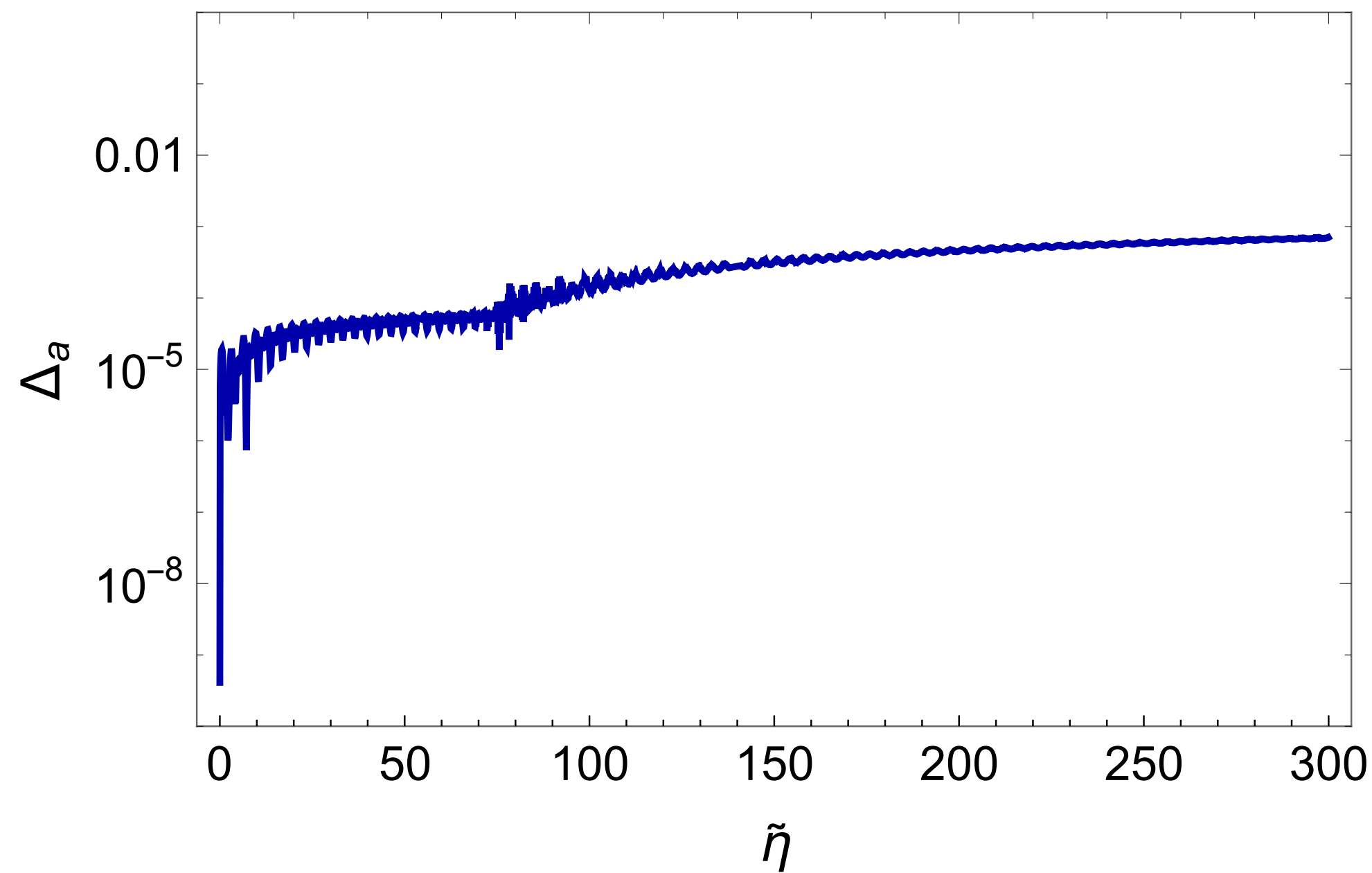


The scale factor behaves as **radiation-dominated** during the entire post-inflationary history



Output: energy conservation

- Algorithms use **second Friedmann equation** to **evolve the scale factor**.
- The **first Friedmann equation** can be used to check the accuracy of the simulation. We denote this “energy conservation”.



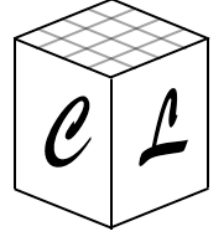
$$a'^2 = \frac{a^{2\alpha+2}}{3} \left(\frac{f_*}{m_p} \right)^2 \langle \tilde{\rho} \rangle$$



$$\Delta_a \equiv \frac{\langle \text{LHS} - \text{RHS} \rangle}{\langle \text{LHS} + \text{RHS} \rangle}$$

(we must have $\Delta_a \ll 1$)

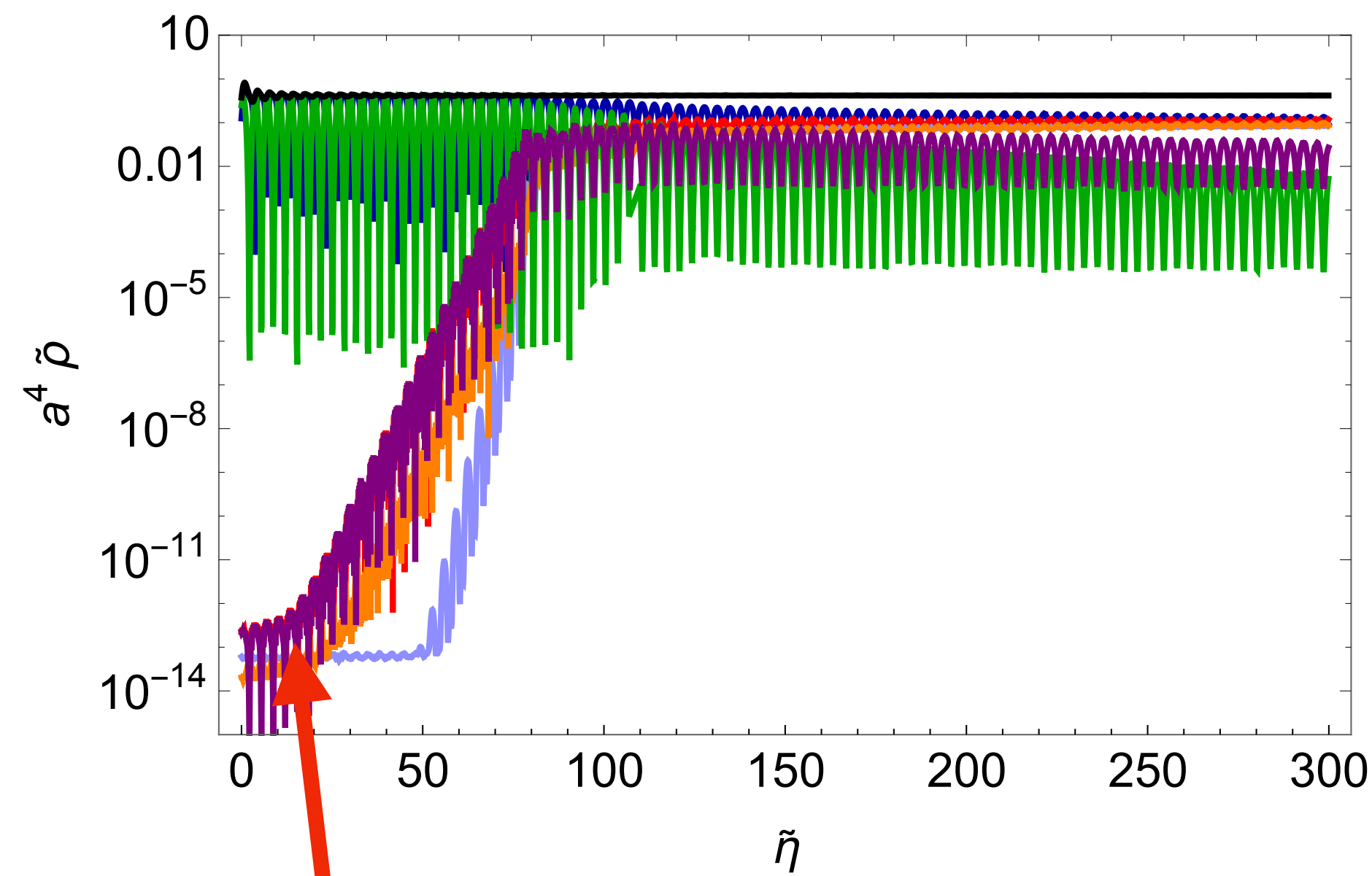
**GOOD
ENERGY
CONSERVATION!**



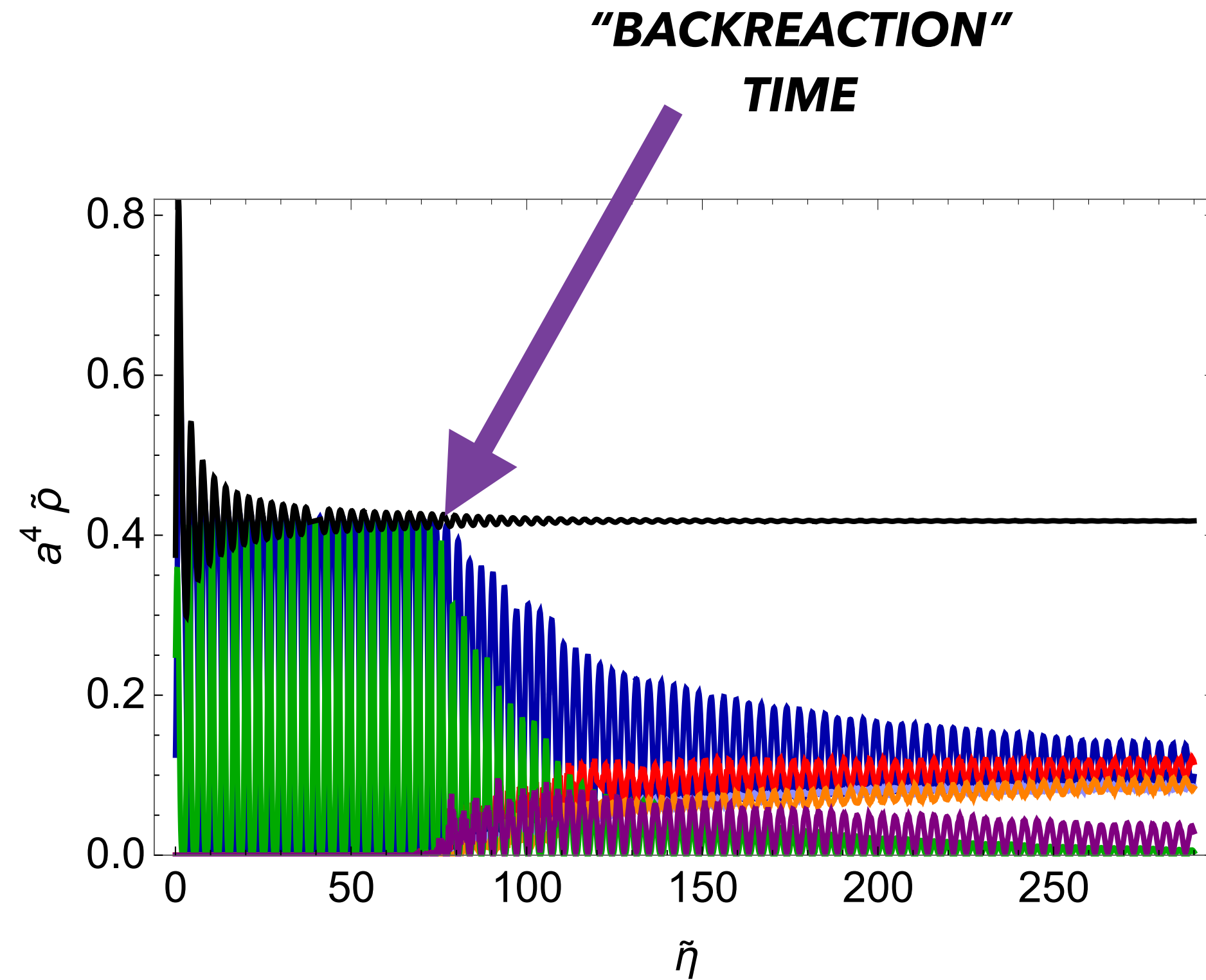
Output: energy components

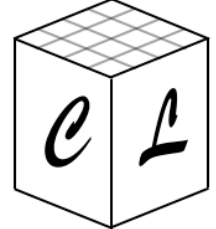
$$\langle \tilde{\rho} \rangle \equiv \frac{\langle \rho \rangle}{f_*^2 \omega_*^2} = \left(\tilde{E}_K^{(0)} + \tilde{E}_V^{(0)} + \tilde{E}_G^{(0)} + \tilde{E}_K^{(1)} + \tilde{E}_G^{(1)} + \tilde{E}_V^{(1)} \right)$$

Kinetic inflaton Potential inflaton Gradient inflaton Kinetic daughter Gradient daughter Interaction energy



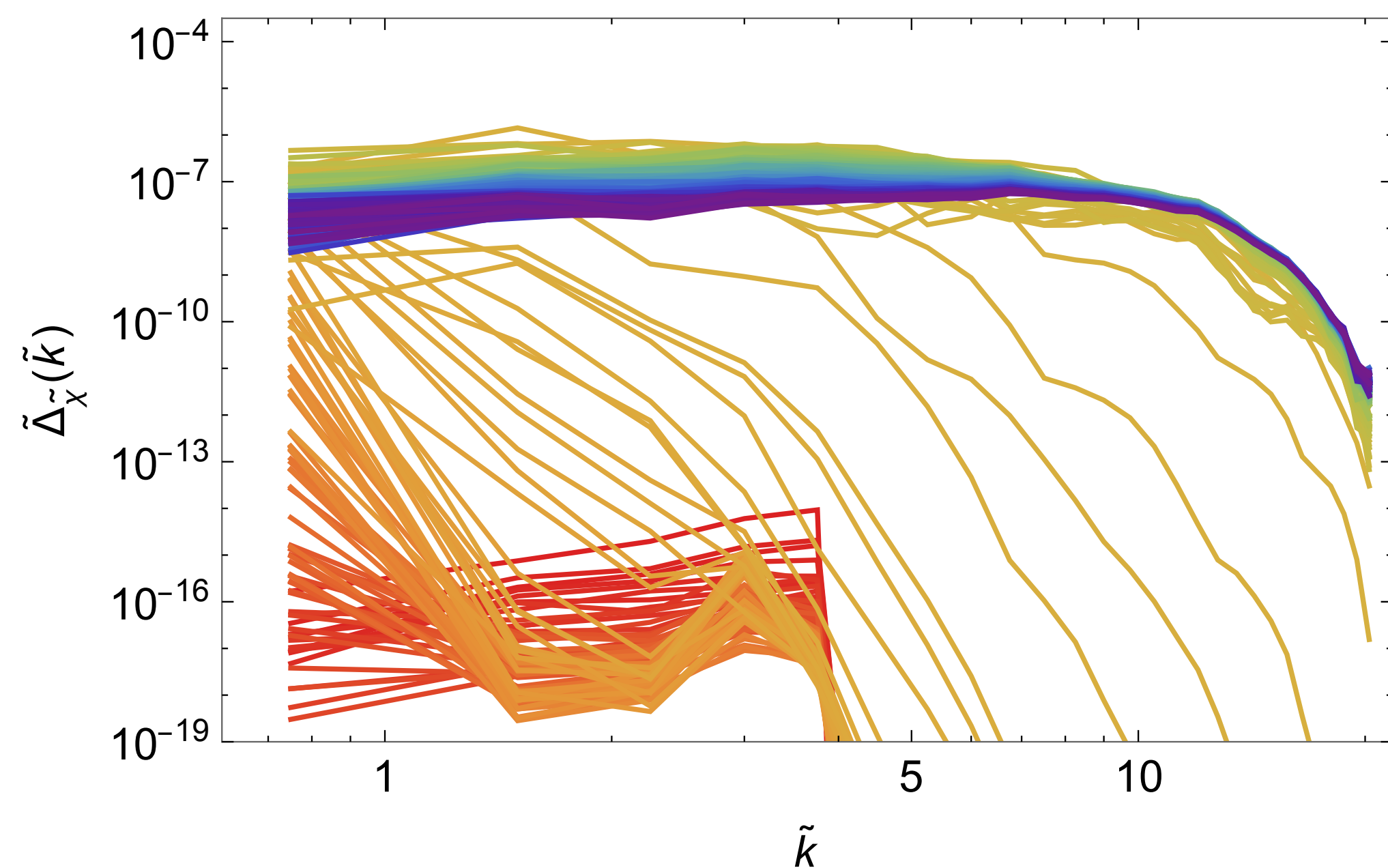
**EXPONENTIAL
GROWTH**



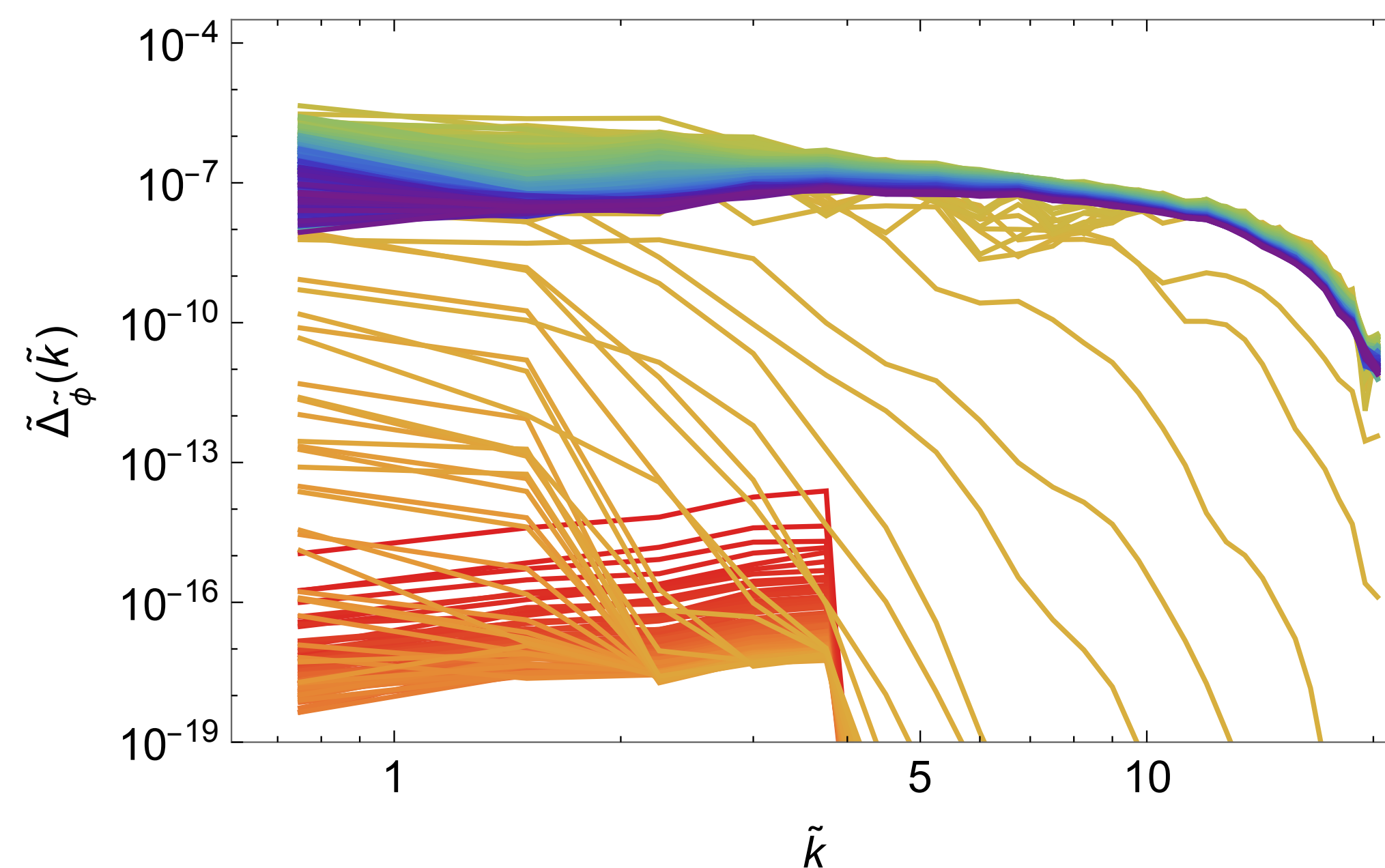


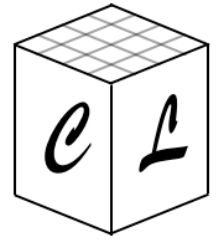
Output: field spectra

Inflaton Spectrum: $\tilde{\Delta}_{\tilde{\phi}} \equiv \frac{\Delta_{\phi}}{f_*^2}$



Daughter Field Spectrum: $\tilde{\Delta}_{\tilde{X}} \equiv \frac{\Delta_X}{f_*^2}$





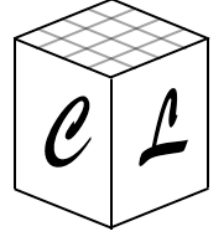
Exercise 1: Let's practice

1. Run the **lphi4** model with the default parameters (the ones provided in the lphi4.in input file). Plot figures for the volume-averaged field amplitudes and variances, energy components, and field spectra. You can use the provided Mathematica or Python notebooks.

Lattice parameters:	$N = 32$	Model parameters:	$q \equiv g^2/\lambda = 100$
	$\delta\tilde{\eta} = 0.01$		$\lambda = 9 \cdot 10^{-14}$
	$\tilde{k}_{\text{IR}} = 0.75$		

Compile and run:

```
cd cosmolattice
cd build
cmake -DMODEL=lphi4 ../
make cosmolattice
../lphi4 input=../src/models/parameter-files/lphi4.in
```



More exercises

2. Run the **lphi4** model with the same parameters as in exercise 1, but changing the resonance parameter to i) $q \equiv g^2/\lambda = 5$; and ii) $q = 0$. Plot the field spectra and observe the differences with respect to exercise 1 (where $q = 100$).
3. Modify the **lphi4** model to add a second daughter field coupled to the inflaton:

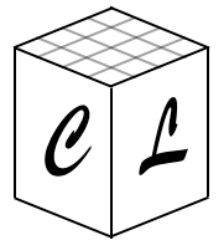
$$V(\phi) = \frac{1}{4}\lambda\phi^4 + \frac{1}{2}g_1^2\phi^2X_1^2 + \frac{1}{2}g_2^2\phi^2X_2^2$$

Run the model for parameters:

$$q_1 \equiv \frac{g_1^2}{\lambda} = 10^2 \quad q_2 \equiv \frac{g_2^2}{\lambda} = 10^2 \quad \lambda = 9 \cdot 10^{-14}$$

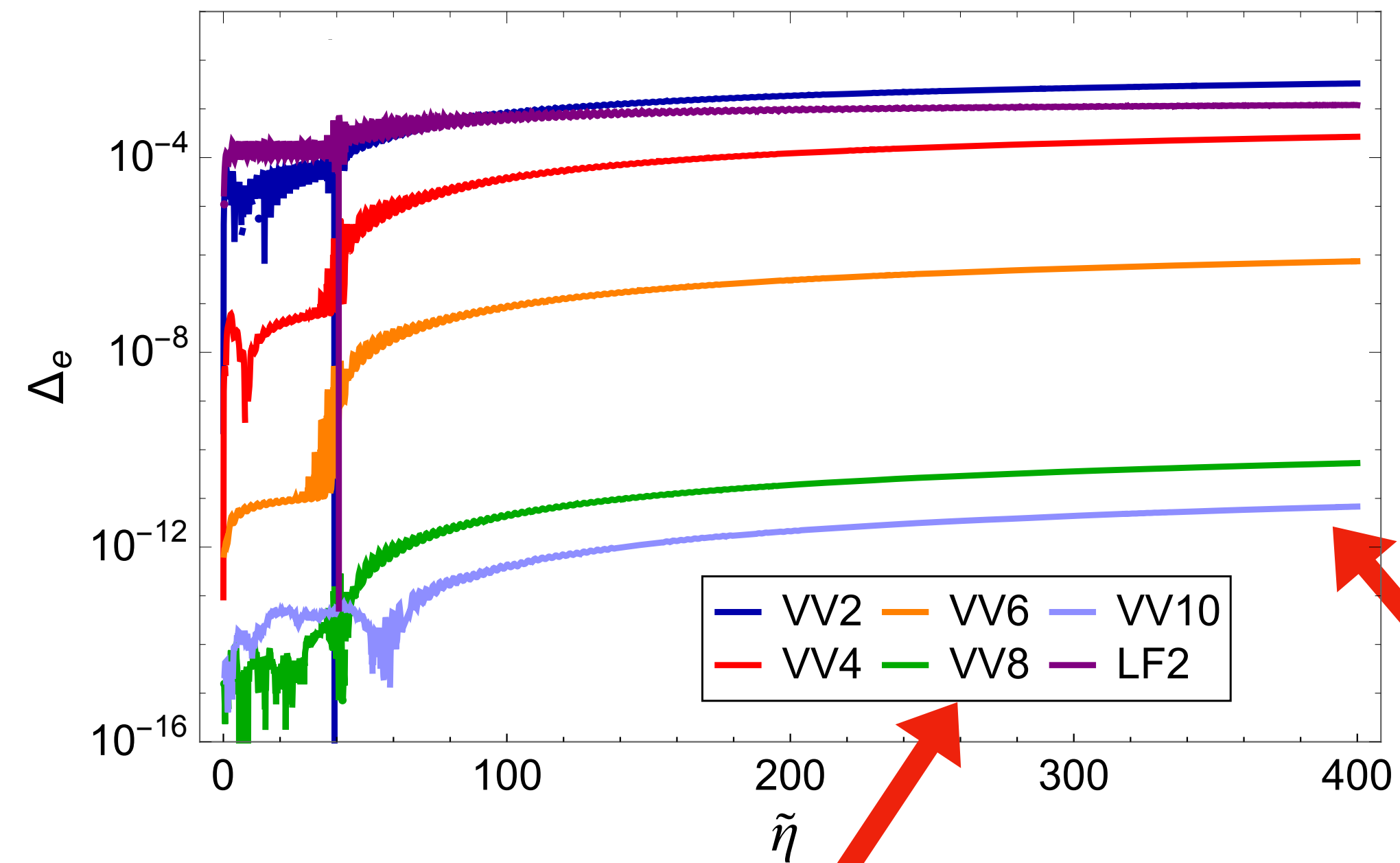
4. Run the **lphi4** model with the default parameters (lphi4.in), but:
 - a) Change the evolution algorithm from VV2 to VV4.
 - b) Reduce the time step by half.

How does the energy conservation improve in each of the two cases with respect to the simulation of exercise 1. Which of the two simulations is faster? Note: The simulation time is given in the info file.



Digression: energy conservation and accuracy of evolver

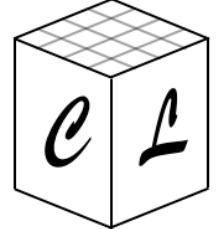
- Energy conservation can be improved if we increase the order of accuracy of the simulation:



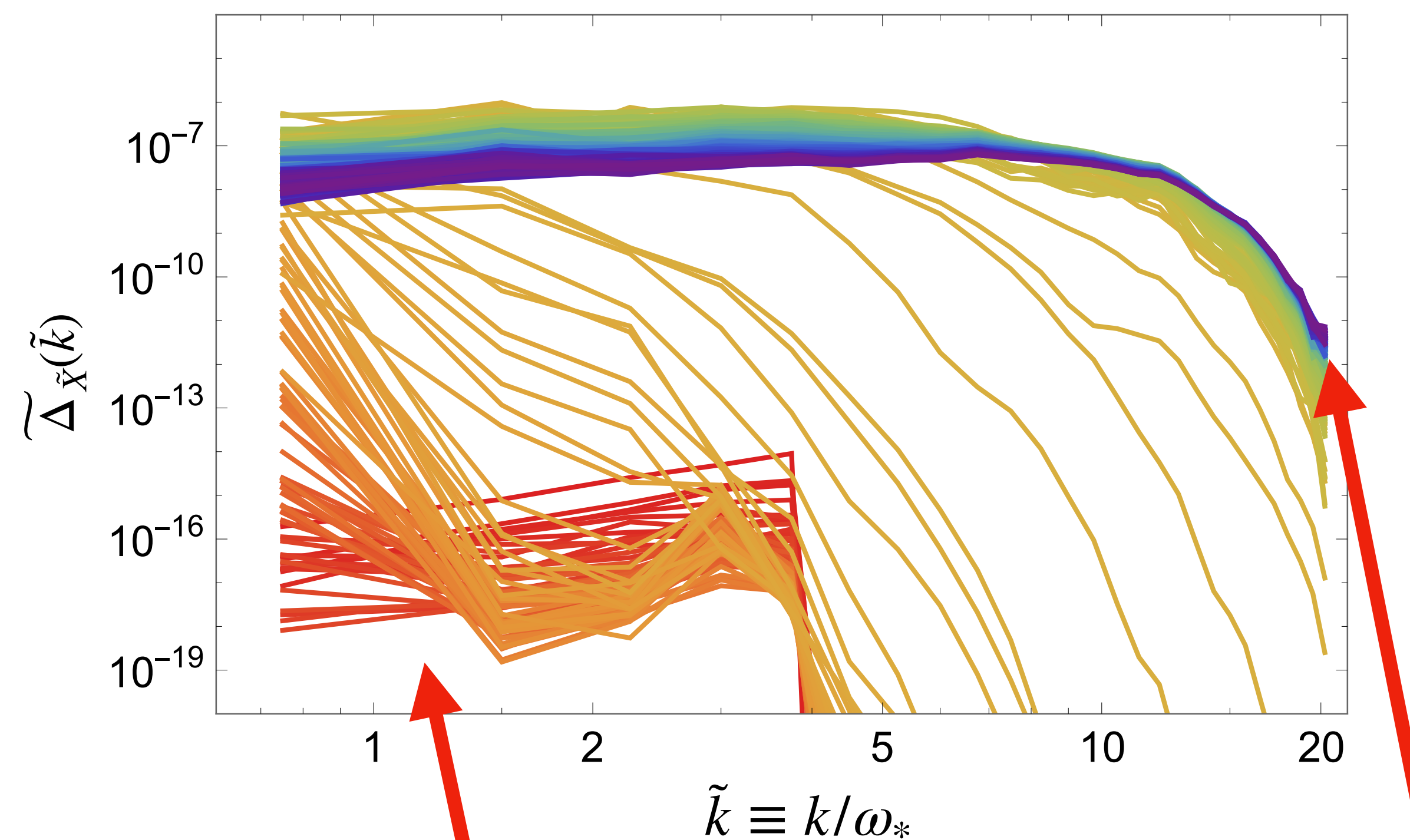
Evolution algorithms:

- **VVN**: Velocity-verlet of accuracy order $O(dt^n)$
- **LF2**: Staggered leapfrog, accuracy order $O(dt^2)$

Energy conserved
up to machine
precision for VV10!

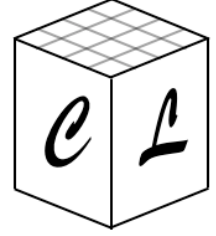


Lattice parameters and IR/UV coverage



BAD IR COVERAGE!
Dominant resonance band not
well captured

Good UV coverage:
UV tail (generated during
the non-linear regime) well
captured



Lattice parameters and IR/UV coverage

► **Infrared cutoff:** $\tilde{k}_{\text{IR}} = \frac{2\pi}{\tilde{L}} = \frac{2\pi}{N\delta\tilde{x}}$ (Note: $\tilde{L} = N\delta\tilde{x}$ is indicated above the denominator)

► **Ultraviolet cutoff:** $\tilde{k}_{\text{max}} = \sqrt{3}\frac{N}{2}\tilde{k}_{\text{IR}} = \frac{\pi}{\delta\tilde{x}}$

• **In our simulation:**

$$N = 32 \quad \tilde{k}_{\text{IR}} = 0.75 \quad \delta\tilde{\eta} = 0.01$$

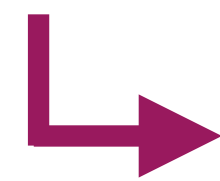
↓

$$\tilde{k}_{\text{max}} = 20.8 \quad \delta\tilde{x} = 0.26$$

We want to decrease $\tilde{k}_{\text{IR}}$. There are two ways:

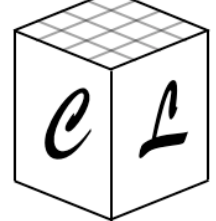
1) **Increase N** (but increases simulation time as $\sim N^3$!)

2) **Increase $\delta\tilde{x}$** (but reduces $\tilde{k}_{\text{UV}}$)



Time step must also be reduced to preserve the Courant stability condition:

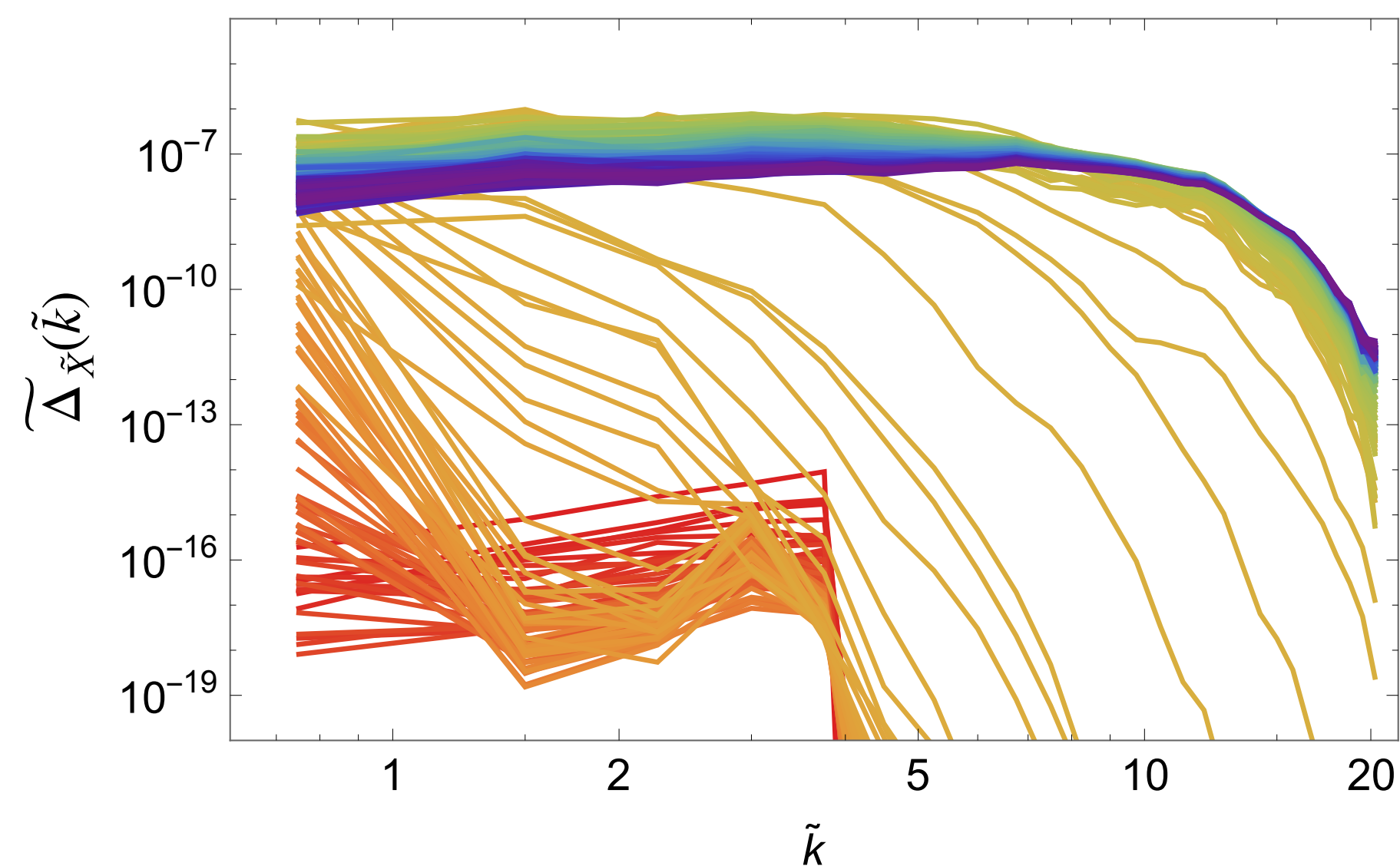
$$\frac{\delta\tilde{\eta}}{\delta\tilde{x}} < \frac{1}{\sqrt{N_{\text{dms}}}}$$



Lattice parameters and IR/UV coverage

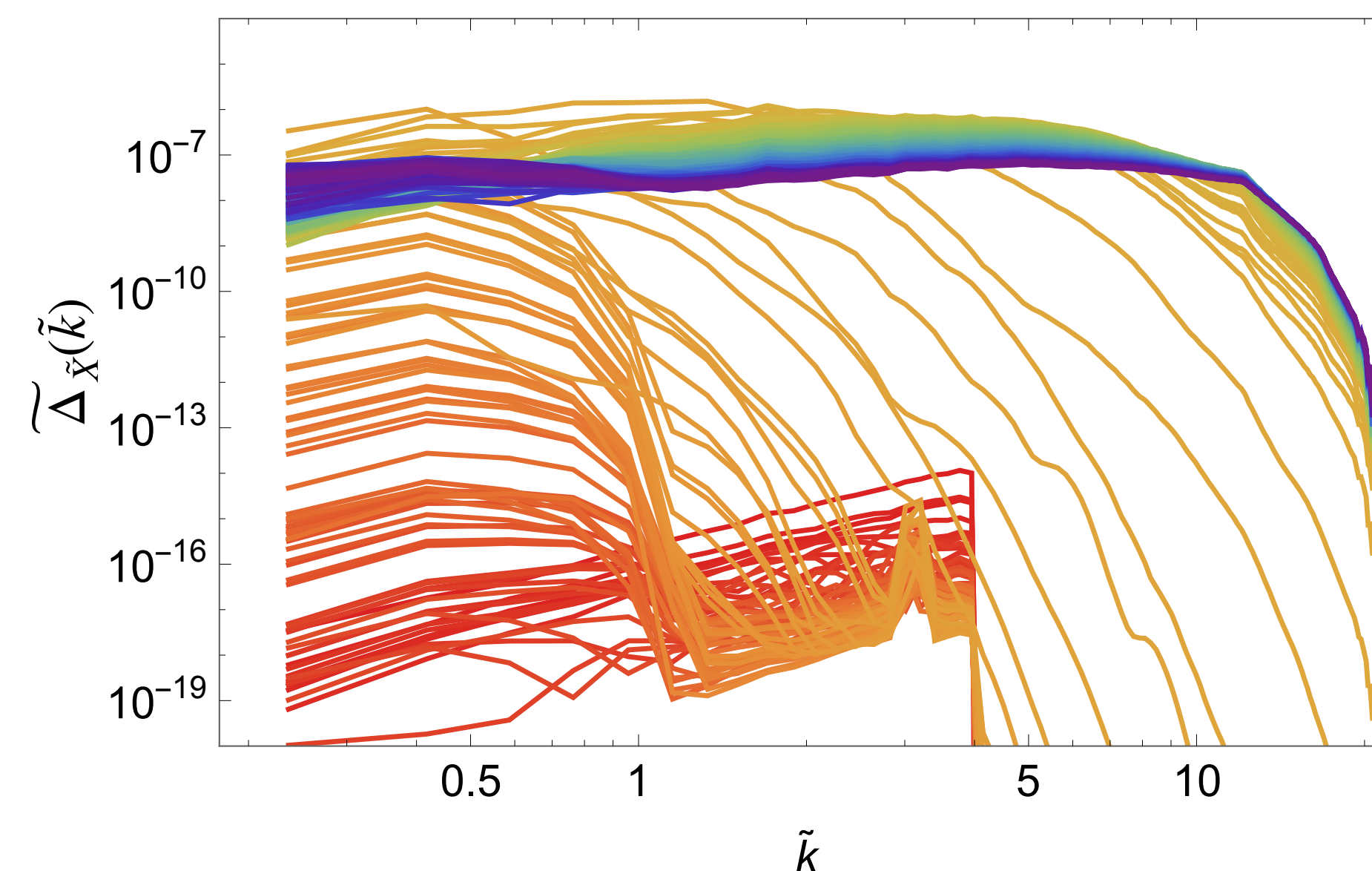
1) Increase **N**:

$N = 32$ $\delta\tilde{x} = 0.26$ $\rightarrow$ $\tilde{k}_{\text{IR}} = 0.75$

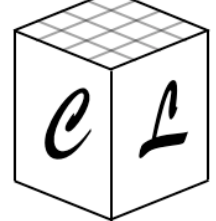


The simulation started on the d3 m7 y2022 around h11m25.
It will be running on a grid of (4,1,1) cores. The simulation finished on the d3 m7 y2022 around h11m25.
The timer indicates that it ran for **17.539s**.
As it ran on 4 cores, this corresponds to 0.0194878 core hours.

$N = 128$ $\delta\tilde{x} = 0.26$ $\rightarrow$ $\tilde{k}_{\text{IR}} = 0.188$



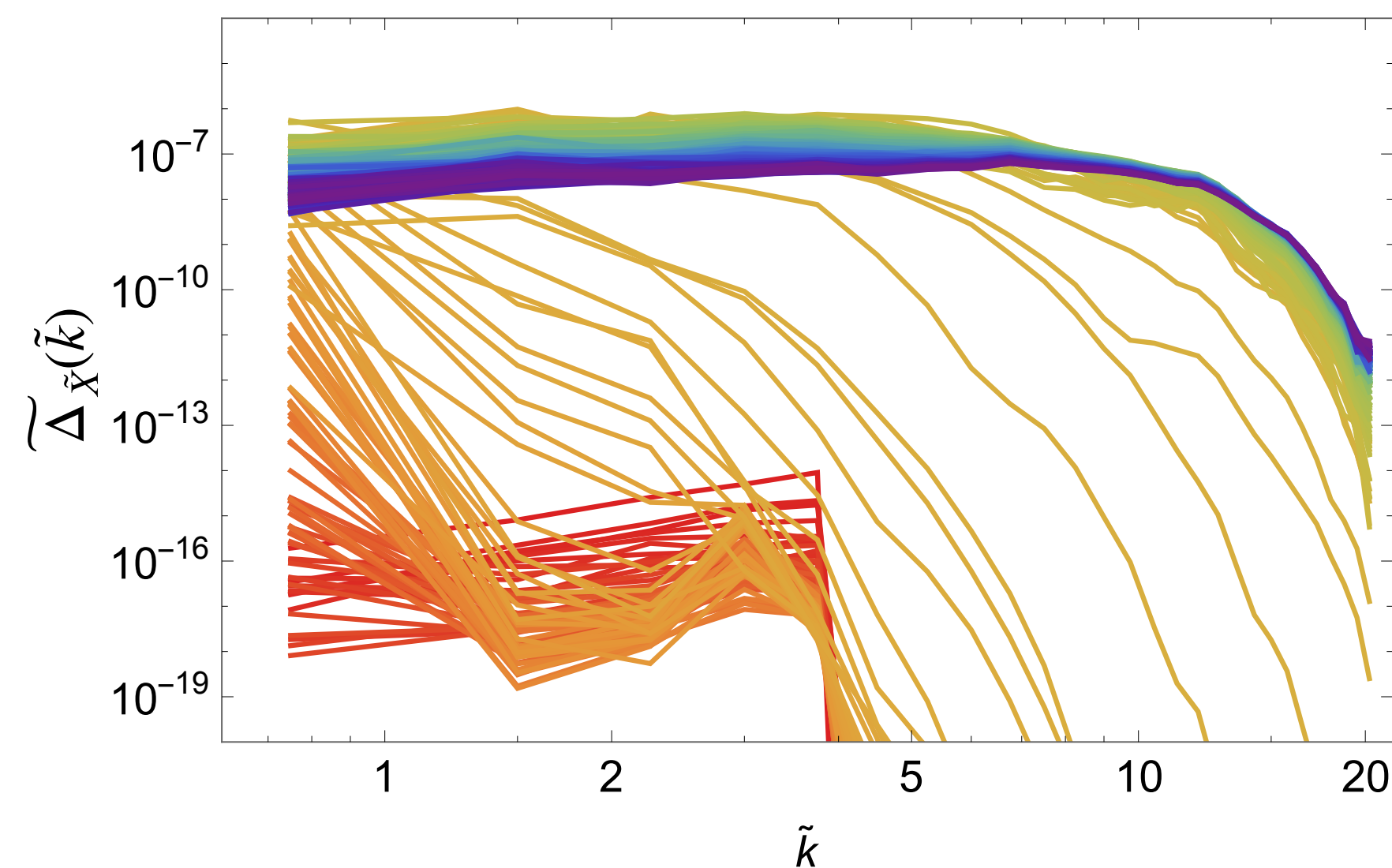
The simulation started on the d3 m7 y2022 around h11m13.
It will be running on a grid of (4,1,1) cores. The simulation finished on the d3 m7 y2022 around h11m25.
The timer indicates that it ran for **697.455s**.
As it ran on 4 cores, this corresponds to 0.77495 core hours.



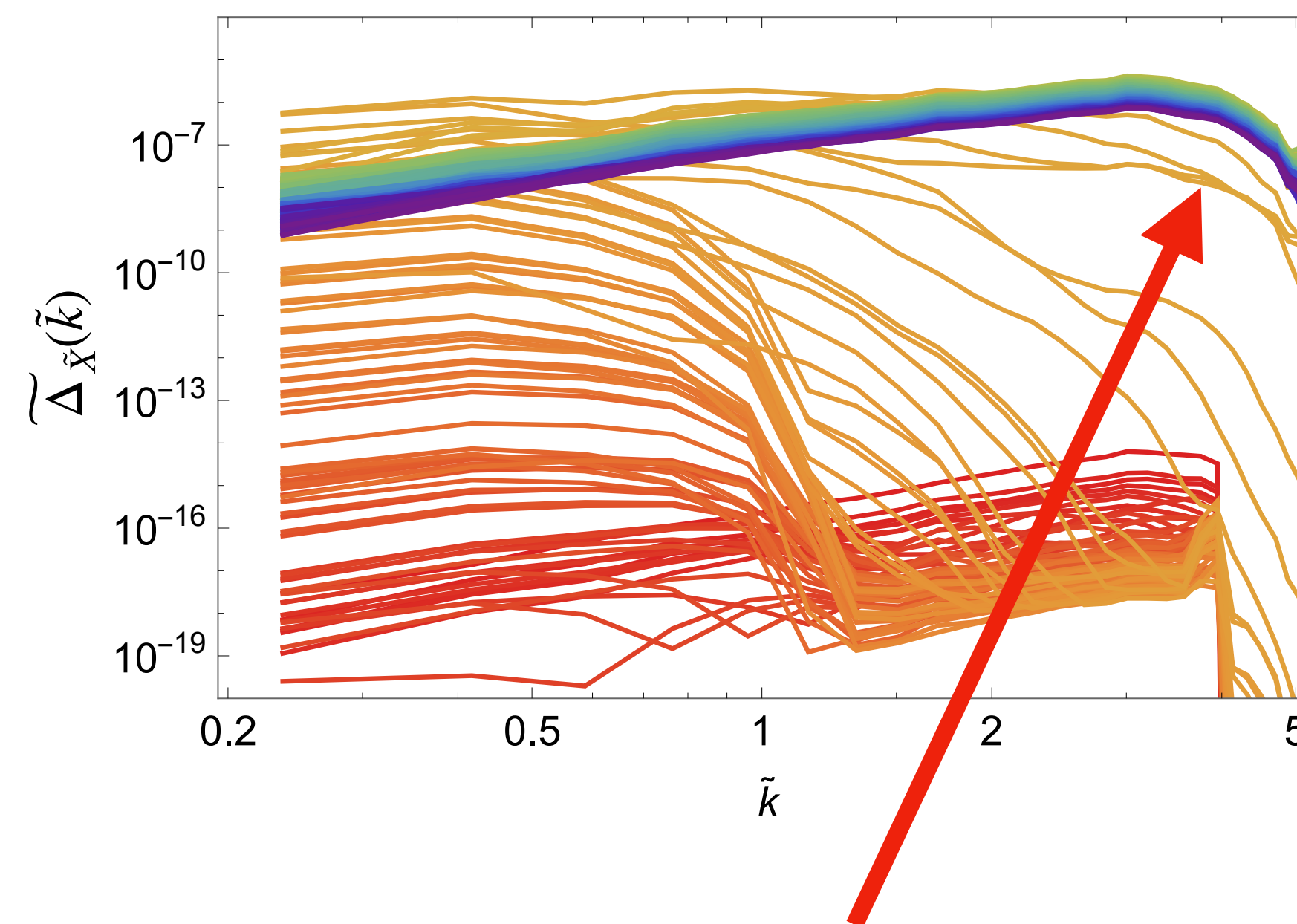
Lattice parameters and IR/UV coverage

2) Increase $\delta\tilde{x}$:

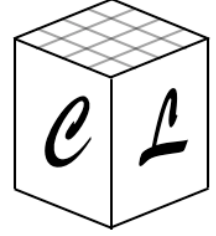
$$N = 32 \quad \begin{matrix} \delta\tilde{\eta} = 0.01 \\ \delta\tilde{x} = 0.26 \end{matrix} \rightarrow \begin{matrix} \tilde{k}_{\text{IR}} = 0.75 \\ \tilde{k}_{\text{max}} = 20.8 \end{matrix}$$



$$N = 32 \quad \begin{matrix} \delta\tilde{\eta} = 0.0025 \\ \delta\tilde{x} = 1.04 \end{matrix} \rightarrow \begin{matrix} \tilde{k}_{\text{IR}} = 0.188 \\ \tilde{k}_{\text{max}} = 5.2 \end{matrix}$$



Simulation time does not increase! BUT: BAD UV COVERAGE!



More exercises

2. Run the **lphi4** model with the same parameters as in exercise 1, but changing the resonance parameter to i) $q \equiv g^2/\lambda = 5$; and ii) $q = 0$. Plot the field spectra and observe the differences with respect to exercise 1 (where $q = 100$).
3. Modify the **lphi4** model to add a second daughter field coupled to the inflaton:

$$V(\phi) = \frac{1}{4}\lambda\phi^4 + \frac{1}{2}g_1^2\phi^2X_1^2 + \frac{1}{2}g_2^2\phi^2X_2^2$$

Run the model for parameters: $q_1 \equiv \frac{g_1^2}{\lambda} = 10^2$ $q_2 \equiv \frac{g_2^2}{\lambda} = 10^2$ $\lambda = 9 \cdot 10^{-14}$

4. Run the **lphi4** model with the default parameters (lphi4.in), but:
 - a) Change the evolution algorithm from VV2 to VV4.
 - b) Reduce the time step by half.

How does the energy conservation improve in each of the two cases with respect to the simulation of exercise 1. Which of the two simulations is faster? Note: The simulation time is given in the info file.

Thank you!