The Art of Lattice and Gravity Waves from Preheating

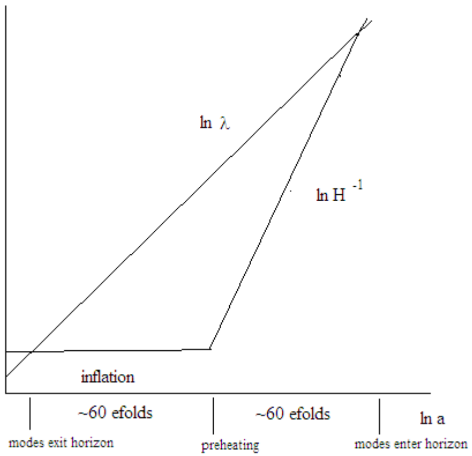
The 3rd IPhT/LPT COSMO meeting

Zhiqi Huang
IPhT, CEA/Saclay

Dec 13, 2010

Outline

Inflation and Preheating



GW from Preheating

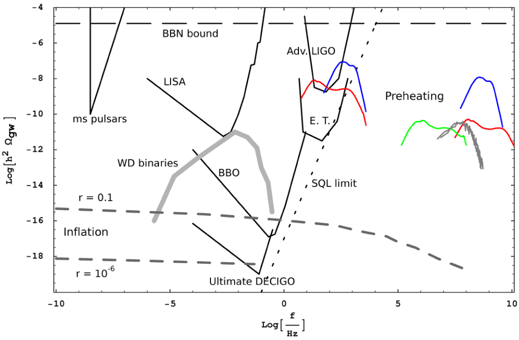


figure source: arXiv:0812.2917 (Dufaux, Felder, Kofman & Navros)

Tachyonic Preheating after Hybrid Inflation

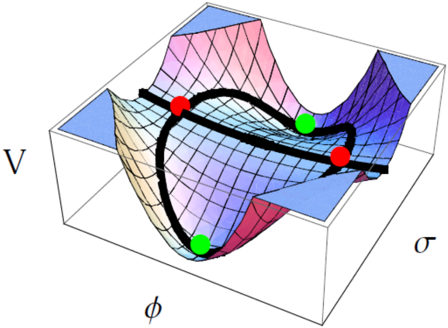


figure source: Dufaux, Felder, Kofman & Navros 2008

ϕ: Inflaton

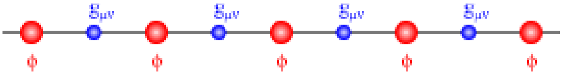
σ: a complex field ($= \sigma_1 + \imath\sigma_2$), it acquires a tachyonic mass ($m^2 < 0$) when ϕ rolls along the ridge.

$$V(\phi, \sigma) = \frac{\lambda}{4}(\sigma^2 - \upsilon^2)^2 + \frac{1}{2}g^2\phi^2\sigma^2 + V_{\rm inf}(\phi)$$

The Wisdom of Discretization

The (1st order) discrete derivatives $\frac{\partial \phi}{\partial x_i}$ are defined on "displaced grids", and so is $g_{\mu\nu}$.

1D example (actually done in 3D with more a sophisticated scheme):



- Discrete differential operators are commutable $\partial_i \partial_j = \partial_j \partial_i$. Operators such as ∇^{-2} are all well defined, so are the discrete scalar, vector, and tensor modes.
- (Discrete) scalar/vector/tensor terms in $\delta T^{\mu\nu}$ only excite (discrete) scalar/vector/tensor metric perturbations.
- ∇^{-2} is NOT equivalent to: discrete Fourier transformation & multiply by $(-k^{-2})$ & inverse discrete Fourier transformation. Need more sophisticated scheme to get traceless transverse (TT) component.

The Wisdom of Discretization

For $k_{\text{min}} \ll k \ll k_{\text{max}}$ the boundary effect and finite resolution effect should disappear. Why sophistications? Why not use projection in Fourier space to get GW (as done in all previous works)?
Two reasons:

- Realistic simulations have $k_{\text{max}} \sim 10^2\text{-}10^3 k_{\text{min}}$. So $k_{\text{min}} \ll k \ll k_{\text{max}}$ is never that well satisfied.
- Scalar metric perturbations $\gg$ tensor metric perturbations. A tiny "leaking" from $g_{\mu\nu,\text{scalar}}$ to $g_{\mu\nu,\text{tensor}}$ can significantly change the GW spectrum.

Evolving A Classical System on the Computer

A trivial system the computer can evolve “exactly”: a free particle with mass m.
Hamiltonian $H = \frac{p^2}{2m}$.

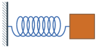
Evolving from t to $t + \Delta t$:

$$p(t + \Delta t) = p(t) \ , \tag{1}$$

$$x(t + \Delta t) = x(t) + \frac{p(t)}{m}\Delta t \ . \tag{2}$$

Only round-off errors ($\lesssim 10^{-16}$ with double precision).

Evolving A Classical System on the Computer



A classical harmonic osillator is already challenging:
$H = \frac{p^2}{2} + \frac{q^2}{2}$.

Evolving from t to $t + \Delta t$:

$$p(t+\Delta t) = p(t) - q(t)\Delta t + O(\Delta t^2) \ , \tag{3}$$

$$q(t+\Delta t) = q(t) + p(t)\Delta t + O(\Delta t^2) \tag{4}$$

- Shrinking Δt usually does not help much.
- A better solution is to change the scheme to make the error term $\propto \Delta t^n$. Large n $\Rightarrow$ only round-off errors.

Evolving A Classical System on the Computer

For a general Hamiltonian $H(\mathbf{p}, \mathbf{q})$ and a variable $f(\mathbf{p}, \mathbf{q})$:

$$\frac{df}{dt} = \hat{\mathcal{H}} f, \tag{5}$$

where the Poisson bracket operator $\hat{\mathcal{H}} \equiv \{\cdot, H\}$.

The exact solution can be formally written as

$$f(t + \Delta t) = e^{\hat{\mathcal{H}}\Delta t} f(t) \tag{6}$$

- The computer knows how to calculate $e^{\hat{\mathcal{H}}\Delta t} f$ EXACTLY if $H = T(\mathbf{p}) \equiv \mathbf{p}^2/2$ or $H = V(\mathbf{q})$, but NOT in the case $H = T(\mathbf{p}) + V(\mathbf{q})$.
- This is because $\hat{\mathcal{T}} \equiv \{\cdot, T\}$ and $\hat{\mathcal{V}} \equiv \{\cdot, V\}$ are not commutable:

$$e^{\hat{\mathcal{H}}\Delta t} = e^{(\hat{\mathcal{T}} + \hat{\mathcal{V}})\Delta t} \neq e^{\hat{\mathcal{T}}\Delta t} e^{\hat{\mathcal{V}}\Delta t} \ . \tag{7}$$

Evolving A Classical System on the Computer

The idea of symplectic integrator (SI) is to find constant c-numbers $c_1, c_2, ..., d_1, d_2, ...,$ such that:

$$e^{(\hat{\mathcal{T}}+\hat{\mathcal{V}})\Delta t} = e^{c_1\hat{\mathcal{T}}\Delta t}e^{d_1\hat{\mathcal{V}}\Delta t}e^{c_2\hat{\mathcal{T}}\Delta t}e^{d_2\hat{\mathcal{V}}\Delta t}... + O(\Delta t^{n+1}) \ . \qquad (8)$$

- The computer knows how to use the right-hand-side of Eq. (8) to evolve the system EXACTLY, if the $O(\Delta t^{n+1})$ term is dropped.
- The 2nd order SI ($n = 2$) is the well known "leap frog" algorithm (used in LatticeEasy & DEFROST when expansion of universe is ignored). Its explict form is

$$e^{(\hat{\mathcal{T}}+\hat{\mathcal{V}})\Delta t} = e^{\frac{1}{2}\hat{\mathcal{T}}\Delta t}e^{\hat{\mathcal{V}}\Delta t}e^{\frac{1}{2}\hat{\mathcal{T}}\Delta t} + O(\Delta t^3) \ . \qquad (9)$$

- Up to 8th order SI's are all known.

The Lattice Code

Scalar fields on a lattice:

$$H(\phi, g_{\mu\nu}) = \hat{\mathcal{A}} + \hat{\mathcal{B}} + \hat{\mathcal{C}}, \tag{10}$$

where the three noncommutative terms are
$\hat{\mathcal{A}}$: the diagonal terms in the kinetic energy of gravity.
$\hat{\mathcal{B}}$: the kinetic energy of the fields and the off-diagonal terms in the kinetic energy of gravity.
$\hat{\mathcal{C}}$: the potential and gradient energy of the fields and gravity

- The high-order symplectic factorization of $e^{(\hat{\mathcal{A}}+\hat{\mathcal{B}}+\hat{\mathcal{C}})\Delta t}$ is unknown (iterative expansion $\Rightarrow$ hundreds of factors, practically useless.)
- Operator $\hat{\mathcal{A}}$ itself is non-canonical (i.e., it has some non-splittable terms containing both $p_{\mu\nu}$ and $g_{\mu\nu}$, the computer does not know how to calculate $e^{\hat{\mathcal{A}}\Delta t} f$ exactly.

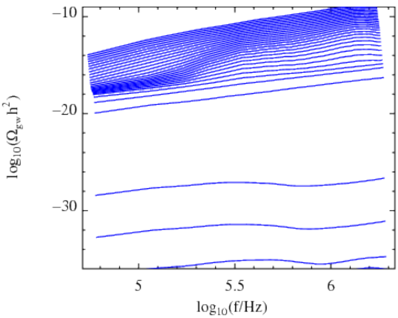
The Lattice Code

What I have done:

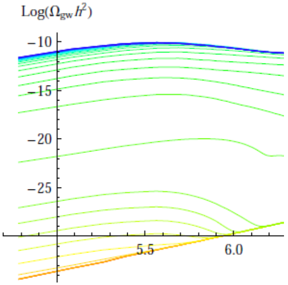
- Find the 4-th and 6-th order optimal symplectic factorizations of $e^{(\hat{\mathcal{A}}+\hat{\mathcal{B}}+\hat{\mathcal{C}})\Delta t}$.
- Use 4-th order Runge-Kutta integrator (with much smaller time-steps) to calculate $e^{\hat{\mathcal{A}}\Delta t}f$ (no cost of extra memory: because $\hat{\mathcal{A}}$ does not contain the gradient terms, I can calculate it grid-wise-ly.)

Features of the code:

- You can choose 2nd, 4th or 6th order symplectic integrators (CPU time cost – 1:3:7. Memory cost: all minimal), depending on how accurate you want your result to be.
- Three modes for the gravity: FRW background with metric perturbations; FRW background without metric perturbations; Minkowski spacetime.



HLattice output: metric perturbation feedbacks; evolution in real space; h^{TT} separated in real space

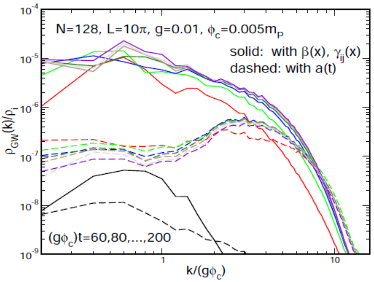


arXiv:0812.2917 (Dufaux, Felder, Kofman & Navros)

LatticeEasy output: no metric perturbation feedbacks; evolution in real space; TT component separated in Fourier space.

The opposite claim?

The opposite claim: non-linear enhancement of GW on large scales?



arXiv:1005.4054 (Bastero-Gil, Macias-Pérez and Santos).

TT component separated in Fourier space; solid/dashed: with/without metric perturbation feedbacks;

leaking from scalar to tensor? numerical noise?

Conclusions

- A discretization scheme allows the separation of scalar, vector and tensor modes.
- Symplectic integrator (with Runge-Kutta sub-integrator for non-canonical terms): evolve the discretized system almost exactly (numerical noise $\sim 10^{-14}$).
- Find lower IR tail of the GW spectrum. Is the extra IR power in the literature from non-perfect discretization?