oscode: fast solutions of oscillatory ODEs in cosmology

Fruzsina Agocs

Astrophysics group, Cavendish Laboratory
Kavli Institute for Cosmology, Cambridge

January 29, 2020

Outline

Motivation

Algorithm

Applications
- Airy and 'burst' equations
- Quantum mechanics
- Cosmology

Extensions

Summary

Motivation

What is oscode and why we need it

Motivation

What is oscode and why we need it

- A numerical solver for **osc**illatory **o**rdinary **d**ifferential **e**quations:

Motivation

What is oscode and why we need it

- A numerical solver for **osc**illatory **o**rdinary **d**ifferential **e**quations:

$$\ddot{x} + 2\gamma(t)\dot{x} + \omega^2(t)x = 0$$

Motivation

What is oscode and why we need it

- A numerical solver for **osc**illatory **o**rdinary **d**ifferential **e**quations:

$$\ddot{x} + 2\gamma(t)\dot{x} + \omega^2(t)x = 0$$

- Even if $\omega(t)$ changes *slowly*, if it's large enough, numerical solution slow

Motivation

What is oscode and why we need it

- A numerical solver for **osc**illatory **o**rdinary **d**ifferential **e**quations:

$$\ddot{x} + 2\gamma(t)\dot{x} + \omega^2(t)x = 0$$

- Even if $\omega(t)$ changes *slowly*, if it's large enough, numerical solution slow
- Conventional (e.g. Runge–Kutta) methods need to step through each peak and trough

Motivation

What is `oscode` and why we need it

- A numerical solver for **osc**illatory **o**rdinary **d**ifferential **e**quations:

$$\ddot{x} + 2\gamma(t)\dot{x} + \omega^2(t)x = 0$$

- Even if $\omega(t)$ changes *slowly*, if it's large enough, numerical solution slow
- Conventional (e.g. Runge–Kutta) methods need to step through each peak and trough
- Number of steps $\propto$ computing time

Motivation

What is oscode and why we need it

- A numerical solver for **osc**illatory **o**rdinary **d**ifferential **e**quations:

$$\ddot{x} + 2\gamma(t)\dot{x} + \omega^2(t)x = 0$$

- Even if $\omega(t)$ changes *slowly*, if it's large enough, numerical solution slow
- Conventional (e.g. Runge–Kutta) methods need to step through each peak and trough
- Number of steps $\propto$ computing time
- Oscillators are extremely common in physics

A brief summary of the algorithm

A brief summary of the algorithm

- Stepping along the numerical solution, initial value problem

A brief summary of the algorithm

- Stepping along the numerical solution, initial value problem
- 2 essential components:

A brief summary of the algorithm

- Stepping along the numerical solution, initial value problem
- 2 essential components:
  1. Switching between approximations: different method for when solution is oscillatory (ω slowly changing) and isn't,

A brief summary of the algorithm

- Stepping along the numerical solution, initial value problem
- 2 essential components:
    1. Switching between approximations: different method for when solution is oscillatory (ω slowly changing) and isn't,
    2. Adaptive stepsize: update stepsize based on error estimate on step and tolerance.

A brief summary of the algorithm

- Stepping along the numerical solution, initial value problem
- 2 essential components:
  1. Switching between approximations: different method for when solution is oscillatory (ω slowly changing) and isn't,
  2. Adaptive stepsize: update stepsize based on error estimate on step and tolerance.
- At each step, attempt to use both methods, and choose one which gives larger stepsize within the given error tolerance

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$
- Represent solution as Taylor-series:

  $$x(t_{i+1}) = x(t_i) + hF_i + \frac{h^2}{2}\frac{dF}{dt}\bigg|_{t_i} + \dots$$

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$
- Represent solution as Taylor-series:
  $x(t_{i+1}) = x(t_i) + hF_i + \frac{h^2}{2}\frac{dF}{dt}\Big|_{t_i} + \dots$
- Higher-order derivatives from evaluations of F:
  $x(t_{i+1}) = x(t_i) + \sum_j b_j hF(x(t_j))$.

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$
- Represent solution as Taylor-series:
  $x(t_{i+1}) = x(t_i) + hF_i + \frac{h^2}{2}\frac{dF}{dt}\Big|_{t_i} + \dots$
- Higher-order derivatives from evaluations of F:
  $x(t_{i+1}) = x(t_i) + \sum_j b_j hF(x(t_j))$.

WKB

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$
- Represent solution as Taylor-series:
  $x(t_{i+1}) = x(t_i) + hF_i + \frac{h^2}{2}\frac{dF}{dt}\Big|_{t_i} + \dots$
- Higher-order derivatives from evaluations of F:
  $x(t_{i+1}) = x(t_i) + \sum_j b_j h F(x(t_j))$.

WKB

- Use the fact that the solution oscillates: $x(t) \sim A(t)e^{i\phi(t)}$

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$
- Represent solution as Taylor-series:
  $x(t_{i+1}) = x(t_i) + hF_i + \frac{h^2}{2}\frac{dF}{dt}\Big|_{t_i} + \dots$
- Higher-order derivatives from evaluations of F:
  $x(t_{i+1}) = x(t_i) + \sum_j b_j hF(x(t_j))$.

WKB

- Use the fact that the solution oscillates: $x(t) \sim A(t)e^{i\phi(t)}$
- $x(t) \sim \frac{1}{\sqrt{\omega(t)}} e^{\pm i \int^t \omega(\tau)d\tau + \dots}$

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Runge–Kutta and Wentzel–Kramers–Brillouin

RK

- $\dot{x} = F(x)$
- Represent solution as Taylor-series:
  $x(t_{i+1}) = x(t_i) + hF_i + \frac{h^2}{2}\frac{dF}{dt}\Big|_{t_i} + \dots$
- Higher-order derivatives from evaluations of F:
  $x(t_{i+1}) = x(t_i) + \sum_j b_j hF(x(t_j))$.

WKB

- Use the fact that the solution oscillates: $x(t) \sim A(t)e^{i\phi(t)}$
- $x(t) \sim \frac{1}{\sqrt{\omega(t)}} e^{\pm i \int^t \omega(\tau)d\tau + \dots}$

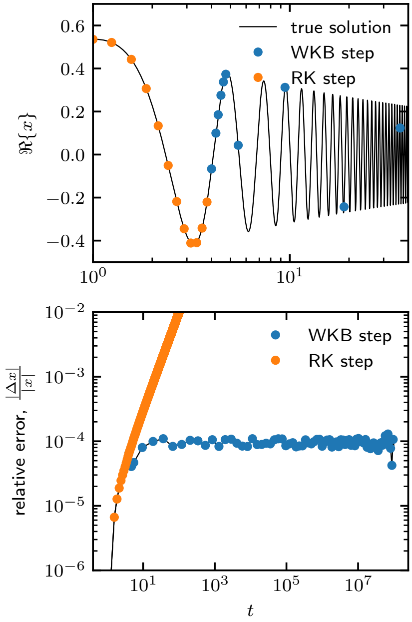
RKWKB[1]

[1] W. J. Handley, A. N. Lasenby, and M. P. Hobson. "The Runge-Kutta-Wentzel-Kramers-Brillouin Method". In: *arXiv e-prints* (Dec. 2016). arXiv: 1612.02288 [physics.comp-ph].

Airy equation

$\ddot{x} + tx = 0$

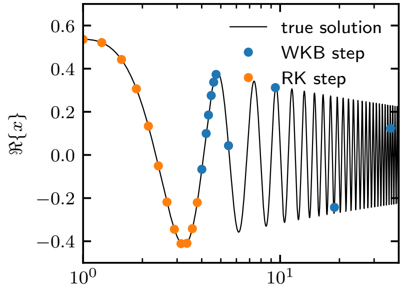
- Maximally hard for RK-based methods, there is analytic solution → study error properties

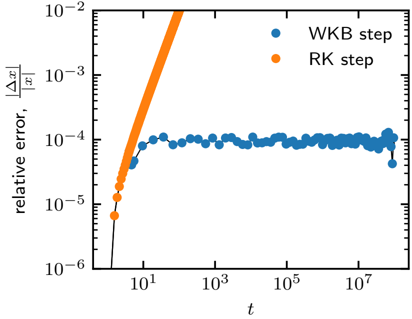


Airy equation

$\ddot{x} + tx = 0$

- Maximally hard for RK-based methods, there is analytic solution → study error properties
- In pure RK, stepsize decreases and numerical error is accumulated

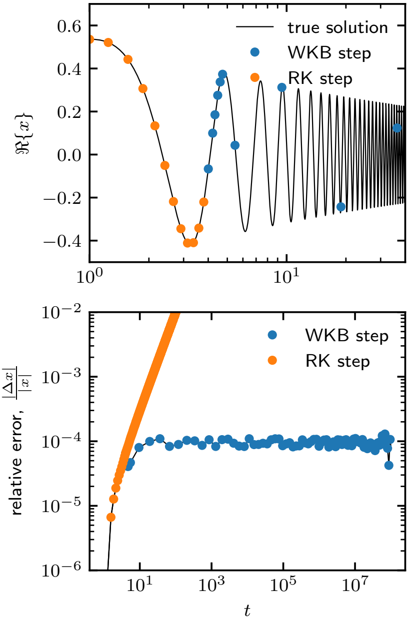




Airy equation

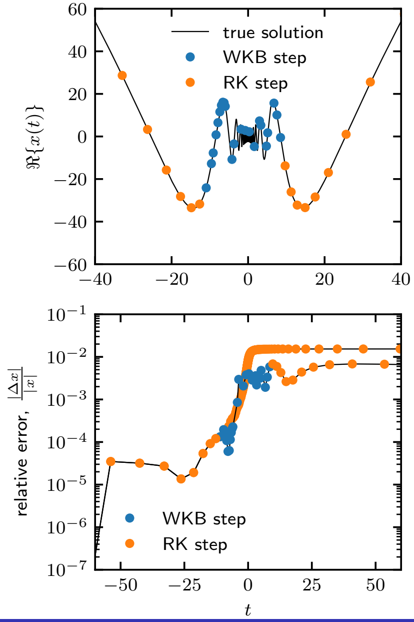
$\ddot{x} + tx = 0$

- Maximally hard for RK-based methods, there is analytic solution → study error properties
- In pure RK, stepsize decreases and numerical error is accumulated
- `oscode` switches from RK to WKB early on, increases stepsize polynomially and stays within error tolerance (10^{-4})



'Burst' equation

$\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$

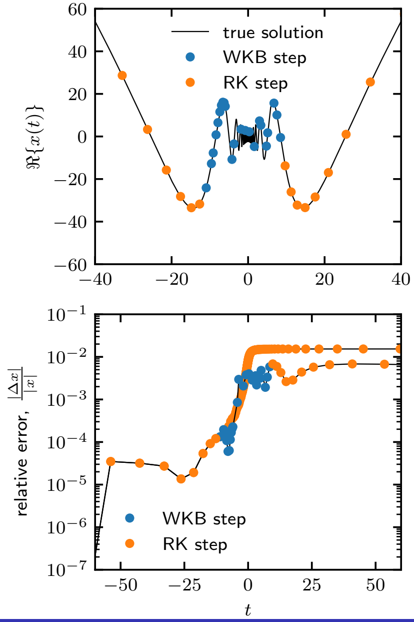


- $\sim n/2$ oscillations within $|t| < n$

'Burst' equation

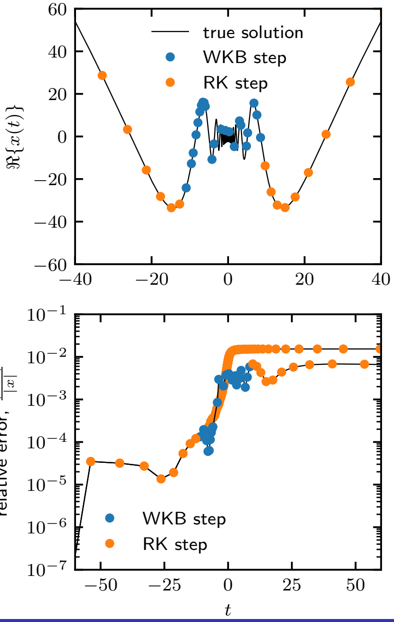
$\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$

- $\sim n/2$ oscillations within $|t| < n$
- $n = 40$ pictured



'Burst' equation

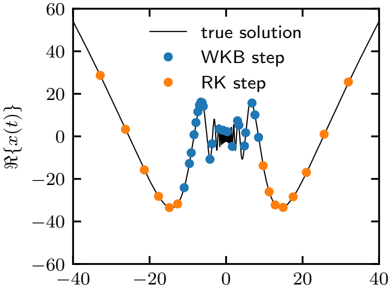
$\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$

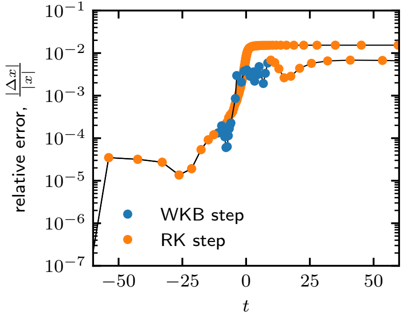


- $\sim n/2$ oscillations within $|t| < n$
- $n = 40$ pictured
- pure RK rapidly accumulates error in central region

'Burst' equation

$\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$



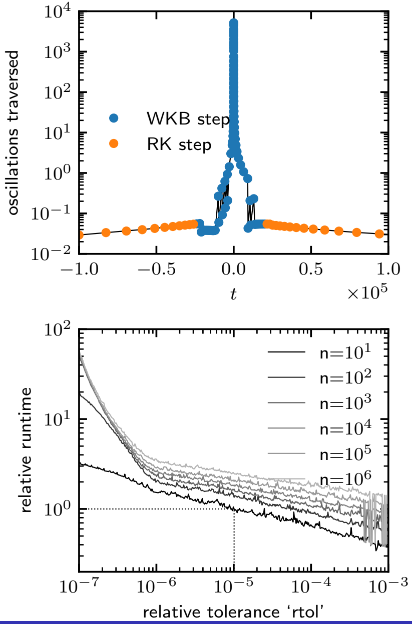


- $\sim n/2$ oscillations within $|t| < n$
- $n = 40$ pictured
- pure RK rapidly accumulates error in central region
- $\sim$ symmetric switching

'Burst' equation

$\ddot{x} + \frac{n^2-1}{(1+t^2)^2} x = 0$

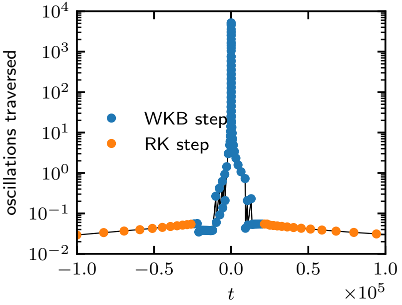
- $n = 10^5$ pictured (but can go up to 10^{10}, then true solution is hard to plot due to sine and cosine's accuracy)

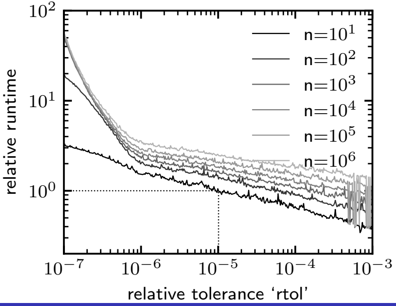


'Burst' equation

$\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$

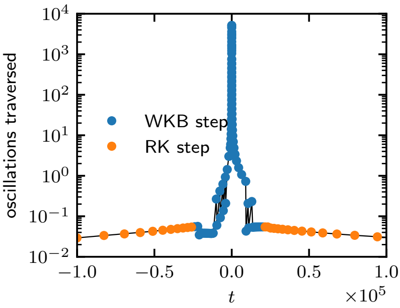
- $n = 10^5$ pictured (but can go up to 10^{10}, then true solution is hard to plot due to sine and cosine's accuracy)
- Number of oscillations traversed in a single step in top figure

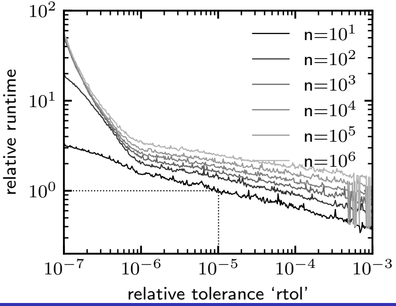




'Burst' equation

$\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$



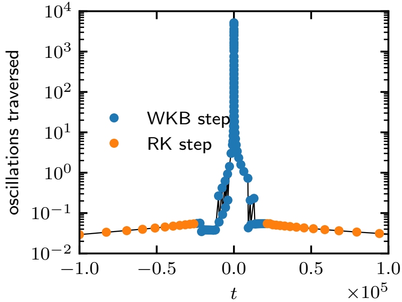


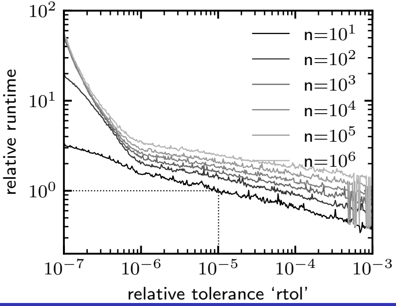
- $n = 10^5$ pictured (but can go up to 10^{10}, then true solution is hard to plot due to sine and cosine's accuracy)
- Number of oscillations traversed in a single step in top figure
- Runtime as function of error tolerance in bottom (relative to $n = 10$, tol= 10^{-5})

'Burst' equation

$\ddot{x} + \frac{n^2-1}{(1+t^2)^2} x = 0$

- $n = 10^5$ pictured (but can go up to 10^{10}, then true solution is hard to plot due to sine and cosine's accuracy)
- Number of oscillations traversed in a single step in top figure
- Runtime as function of error tolerance in bottom (relative to $n = 10$, tol$= 10^{-5}$)
- Gentle scaling of runtime within $10^{-6} <$tol$< 10^{-4}$





Schrödinger equation

$\Psi''(x) + 2m(E - V(x))\Psi(x) = 0$

Can estimate eigenvalues in arbitrary 1D potential:

- Guess eigenvalue E

Schrödinger equation

$\Psi''(x) + 2m(E - V(x))\Psi(x) = 0$

Can estimate eigenvalues in arbitrary 1D potential:

- Guess eigenvalue E
- Start two solutions (ψ_L,ψ_R) from outside the well, initial conditions $\psi = 0$, $\psi' = c$

Schrödinger equation

$\Psi''(x) + 2m(E - V(x))\Psi(x) = 0$

Can estimate eigenvalues in arbitrary 1D potential:

- Guess eigenvalue E
- Start two solutions (ψ_L,ψ_R) from outside the well, initial conditions $\psi = 0$, $\psi' = c$
- If two solutions 'match up' in the middle, eigenvalue is found

Schrödinger equation

$\Psi''(x) + 2m(E - V(x))\Psi(x) = 0$

Can estimate eigenvalues in arbitrary 1D potential:

- Guess eigenvalue E
- Start two solutions (ψ_L,ψ_R) from outside the well, initial conditions $\psi = 0$, $\psi' = c$
- If two solutions 'match up' in the middle, eigenvalue is found
- Minimise $\frac{\Psi'_L}{\Psi_L} - \frac{\Psi'_R}{\Psi_R}$ as a function of the guess E

Schrödinger equation

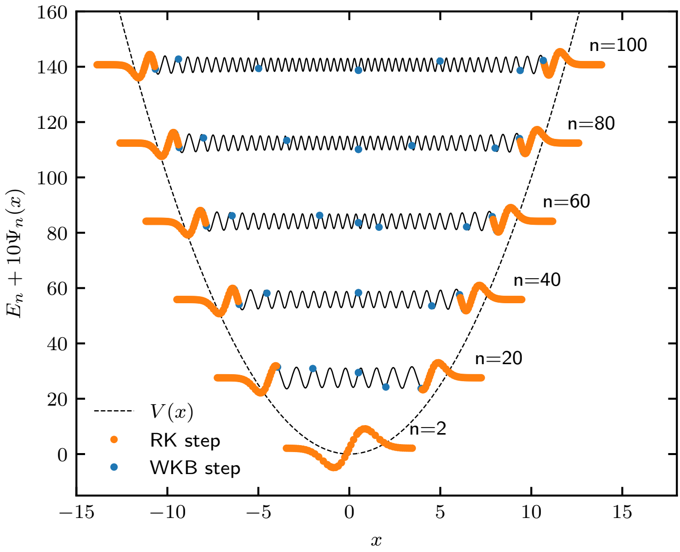
$\Psi''(x) + 2m(E - V(x))\Psi(x) = 0$

Can estimate eigenvalues in arbitrary 1D potential:

- Guess eigenvalue E
- Start two solutions (ψ_L,ψ_R) from outside the well, initial conditions $\psi = 0$, $\psi' = c$
- If two solutions 'match up' in the middle, eigenvalue is found
- Minimise $\frac{\Psi'_L}{\Psi_L} - \frac{\Psi'_R}{\Psi_R}$ as a function of the guess E
- Eigenvalues obtained match reality much more closely than the tolerance set

Harmonic potential well

$V(x) = x^2$



Harmonic well + quartic anharmonicity

$V(x) = x^2 + \lambda x^4$, $\lambda = 1$

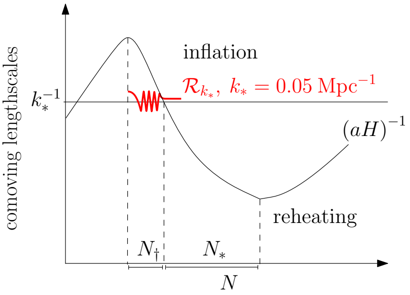
n	E_n^{oscode}	E_n^{*}[2]	$\sim \log_{10} \lvert\Delta E/E\rvert$
0	1.392353	1.392352	-6
1	4.648815	4.648813	-7
2	8.6550501	8.6550500	-8
3	13.156806	13.156804	-7
4	18.0577	18.0576	-5
15	88.6104	88.6103	-6
16	96.1291	96.1296	-5
17	103.793	103.795	-5
18	111.6025	111.6020	-6
19	119.5440	119.5442	-6
50	417.05620	417.05626	-7
100	1035.5440	1035.5442	-7
1000	21932.7848	21932.7840	-8
10000	471103.81	471103.80	-8

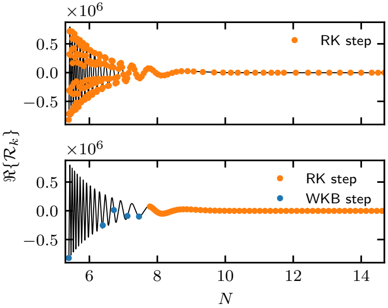
[2] K. Banerjee et al. "The Anharmonic Oscillator". In: *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences* 360.1703 (1978).

Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Governs evolution of curvature perturbation $\mathcal{R}_k$ with lengthscale k^{-1}

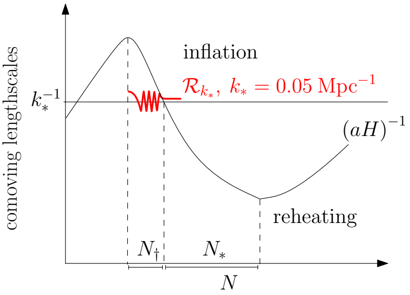


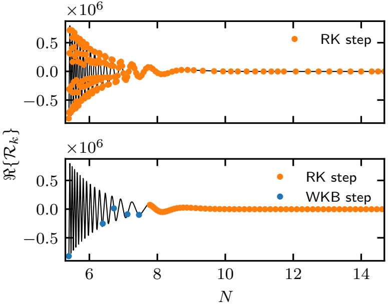


Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Governs evolution of curvature perturbation $\mathcal{R}_k$ with lengthscale k^{-1}
- (Using e-folds of inflation, $N = \ln a$ as independent variable)

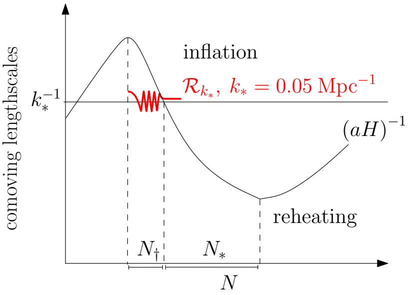


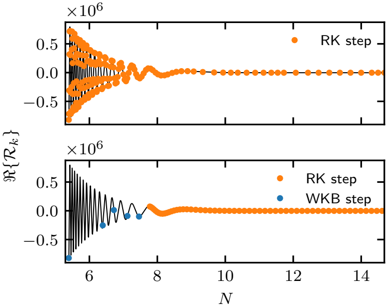


Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Governs evolution of curvature perturbation $\mathcal{R}_k$ with lengthscale k^{-1}
- (Using e-folds of inflation, $N = \ln a$ as independent variable)
- If lengthscale exceeds the comoving Hubble horizon, loss of causal connection $\rightarrow$ 'freeze-out'

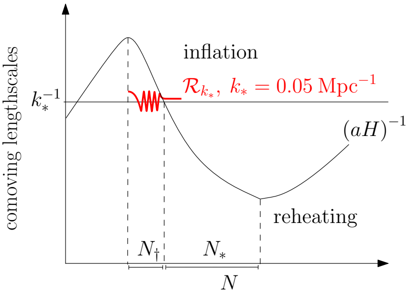


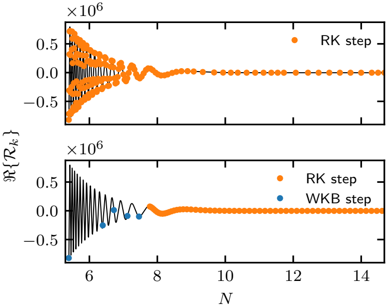


Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Governs evolution of curvature perturbation $\mathcal{R}_k$ with lengthscale k^{-1}
- (Using e-folds of inflation, $N = \ln a$ as independent variable)
- If lengthscale exceeds the comoving Hubble horizon, loss of causal connection → 'freeze-out'
- Power spectrum of $\mathcal{R}_k$ is the primordial power spectrum (PPS), precursor of the CMB





Mukhanov–Sasaki equation

$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$

- Need to compute PPS numerically for many inflationary models, e.g. kinetic dominance[3][4]

[3] L. T. Hergt et al. "Case for kinetically dominated initial conditions for inflation". In: *Phys. Rev. D* 100 (2 July 2019), p. 023502.

[4] L. T. Hergt et al. "Constraining the kinetically dominated universe". In: *Phys. Rev. D* 100 (2 July 2019), p. 023501.

[5] W. I. J. Haddadin and W. J. Handley. *Rapid numerical solutions for the Mukhanov-Sazaki equation*. 2018. arXiv: 1809.11095 [astro-ph.CO].

[6] Will Handley. "Primordial power spectra for curved inflating universes". In: *Phys. Rev. D* 100 (12 Dec. 2019), p. 123517.

Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Need to compute PPS numerically for many inflationary models, e.g. kinetic dominance[3,4]
- But this is challenging at large k

---

[3] L. T. Hergt et al. "Case for kinetically dominated initial conditions for inflation". In: *Phys. Rev. D* 100 (2 July 2019), p. 023502.

[4] L. T. Hergt et al. "Constraining the kinetically dominated universe". In: *Phys. Rev. D* 100 (2 July 2019), p. 023501.

[5] W. I. J. Haddadin and W. J. Handley. *Rapid numerical solutions for the Mukhanov-Sazaki equation*. 2018. arXiv: 1809.11095 [astro-ph.CO].

[6] Will Handley. "Primordial power spectra for curved inflating universes". In: *Phys. Rev. D* 100 (12 Dec. 2019), p. 123517.

Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Need to compute PPS numerically for many inflationary models, e.g. kinetic dominance[3][4]
- But this is challenging at large k
- Other fast solvers exist, but rely on assumptions[5]

---

[3] L. T. Hergt et al. "Case for kinetically dominated initial conditions for inflation". In: *Phys. Rev. D* 100 (2 July 2019), p. 023502.

[4] L. T. Hergt et al. "Constraining the kinetically dominated universe". In: *Phys. Rev. D* 100 (2 July 2019), p. 023501.

[5] W. I. J. Haddadin and W. J. Handley. *Rapid numerical solutions for the Mukhanov-Sazaki equation*. 2018. arXiv: 1809.11095 [astro-ph.CO].

[6] Will Handley. "Primordial power spectra for curved inflating universes". In: *Phys. Rev. D* 100 (12 Dec. 2019), p. 123517.

Mukhanov–Sasaki equation

$$\ddot{\mathcal{R}}_k + 2\left(\frac{\ddot{\phi}}{\dot{\phi}} - \frac{1}{2}\dot{\phi}^2 + \frac{3}{2}\right)\dot{\mathcal{R}}_k + \left(\frac{k}{aH}\right)^2 \mathcal{R}_k = 0$$

- Need to compute PPS numerically for many inflationary models, e.g. kinetic dominance[3][4]
- But this is challenging at large k
- Other fast solvers exist, but rely on assumptions[5]
- Speed up forward-modelling phase of inference significantly (> 1000x), e.g. closed-universe models[6]

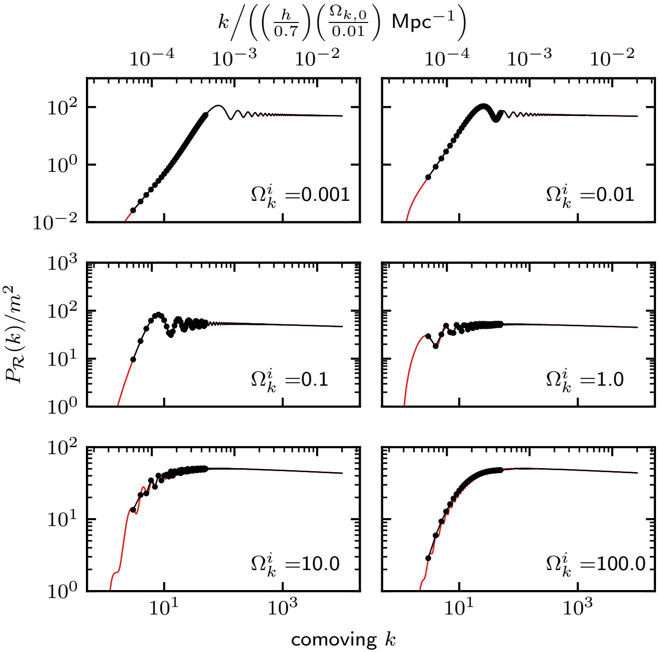
[3] L. T. Hergt et al. "Case for kinetically dominated initial conditions for inflation". In: *Phys. Rev. D* 100 (2 July 2019), p. 023502.

[4] L. T. Hergt et al. "Constraining the kinetically dominated universe". In: *Phys. Rev. D* 100 (2 July 2019), p. 023501.

[5] W. I. J. Haddadin and W. J. Handley. *Rapid numerical solutions for the Mukhanov-Sazaki equation*. 2018. arXiv: 1809.11095 [astro-ph.CO].

[6] Will Handley. "Primordial power spectra for curved inflating universes". In: *Phys. Rev. D* 100 (12 Dec. 2019), p. 123517.

Closed universes



Extensions

- Generalising to many dimensions (is challenging)[7]

[7] Jamie Bamber and Will Handley. "Beyond the Runge-Kutta-Wentzel-Kramers-Brillouin method". In: *arXiv e-prints* (July 2019). arXiv: 1907.11638 [physics.comp-ph].

Extensions

- Generalising to many dimensions (is challenging)[7]
- Generalising to higher order ODEs

[7] Jamie Bamber and Will Handley. "Beyond the Runge-Kutta-Wentzel-Kramers-Brillouin method". In: *arXiv e-prints* (July 2019). arXiv: 1907.11638 [physics.comp-ph].

Extensions

- Generalising to many dimensions (is challenging)[7]
- Generalising to higher order ODEs
- Use an approximation other than WKB

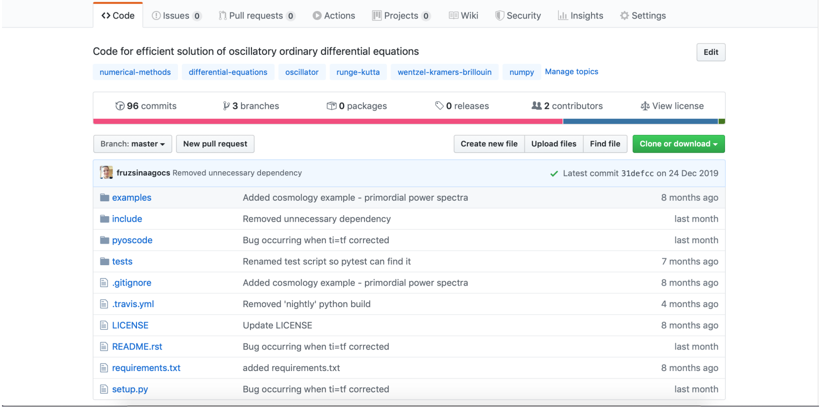
[7] Jamie Bamber and Will Handley. "Beyond the Runge-Kutta-Wentzel-Kramers-Brillouin method". In: *arXiv e-prints* (July 2019). arXiv: 1907.11638 [physics.comp-ph].

Extensions

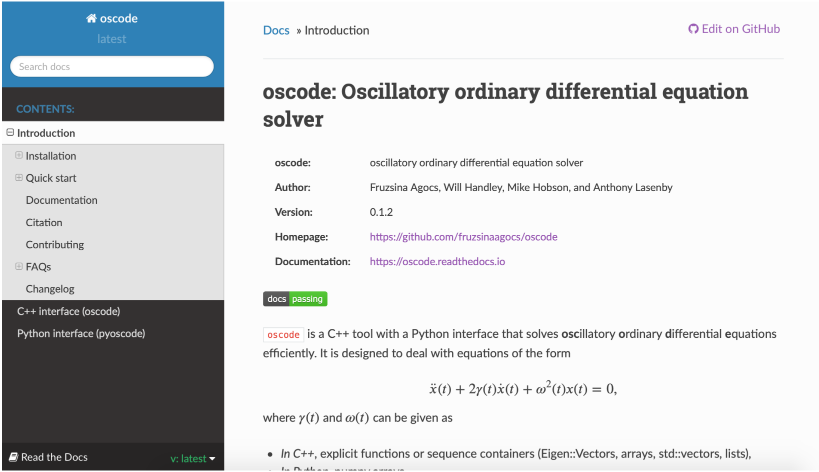
- Generalising to many dimensions (is challenging)[7]
- Generalising to higher order ODEs
- Use an approximation other than WKB
- `oscode` and its underlying algorithm are the beginning of a novel suite of methods

[7] Jamie Bamber and Will Handley. "Beyond the Runge-Kutta-Wentzel-Kramers-Brillouin method". In: *arXiv e-prints* (July 2019). arXiv: 1907.11638 [physics.comp-ph].

Open-source software, documentation, examples



Open-source software, documentation, examples


oscode
latest
Search docs
CONTENTS:
Introduction
Installation
Quick start
Documentation
Citation
Contributing
FAQs
Changelog
C++ interface (oscode)
Python interface (pyoscode)
Read the Docs v: latest

Docs » Introduction
Edit on GitHub

oscode: Oscillatory ordinary differential equation solver

oscode:	oscillatory ordinary differential equation solver
Author:	Fruzsina Agocs, Will Handley, Mike Hobson, and Anthony Lasenby
Version:	0.1.2
Homepage:	https://github.com/fruzsinaagocs/oscode
Documentation:	https://oscode.readthedocs.io

docs passing

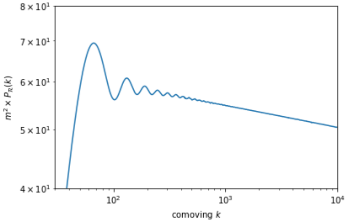
`oscode` is a C++ tool with a Python interface that solves **osc**illatory **o**rdinary **d**ifferential **e**quations efficiently. It is designed to deal with equations of the form

$$\ddot{x}(t) + 2\gamma(t)\dot{x}(t) + \omega^2(t)x(t) = 0,$$

where $\gamma(t)$ and $\omega(t)$ can be given as

- *In C++*, explicit functions or sequence containers (Eigen::Vectors, arrays, std::vectors, lists),

Open-source software, documentation, examples



A closed universe

All we have to do differently is:

1. solve the background equations again with $K = 1$,

In [37]:
```
K = 1
N_i = -1.74
ok_i = 1.0
N = np.linspace(N_i,N_f,Nbg)
# Initial conditions
phi_i = np.sqrt(4.*(1./ok_i + K)*np.exp(-2.0*N_i)/m**2)
logok_i = np.log(ok_i)
y_i = np.array([logok_i, phi_i])
# Solve for the background until the end of inflation
endinfl.terminal = True
endinfl.direction = 1
bgsol = solve_ivp(bgeqs, (N_i,N_f), y_i, events=endinfl, t_eval=N, rtol=1
e-8, atol=1e-10)
```

Number of e-folds of inflation now is

Summary

- `oscode` is a numerical solver for oscillatory ordinary differential equations[8]

[8] F. J. Agocs et al. "Efficient method for solving highly oscillatory ordinary differential equations with applications to physical systems". In: *Phys. Rev. Research* 2 (1 Jan. 2020), p. 013030.

Summary

- `oscode` is a numerical solver for oscillatory ordinary differential equations[8]
- Underlying algorithm switches between methods depending on whether solution is oscillatory

[8] F. J. Agocs et al. "Efficient method for solving highly oscillatory ordinary differential equations with applications to physical systems". In: *Phys. Rev. Research* 2 (1 Jan. 2020), p. 013030.

Summary

- `oscode` is a numerical solver for oscillatory ordinary differential equations[8]
- Underlying algorithm switches between methods depending on whether solution is oscillatory
- Can skip over large regions of oscillations, reducing computation time, speeding up forward modelling

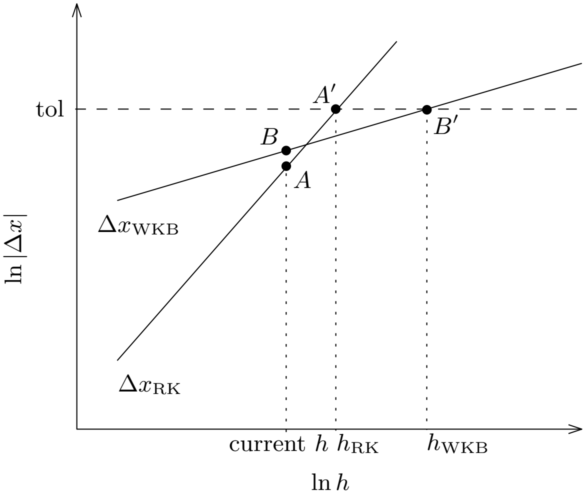
[8] F. J. Agocs et al. "Efficient method for solving highly oscillatory ordinary differential equations with applications to physical systems". In: *Phys. Rev. Research* 2 (1 Jan. 2020), p. 013030.

Summary

- oscode is a numerical solver for oscillatory ordinary differential equations[8]
- Underlying algorithm switches between methods depending on whether solution is oscillatory
- Can skip over large regions of oscillations, reducing computation time, speeding up forward modelling
- Wide range of uses: quantum mechanics, electrical circuits, cosmology, ...

[8] F. J. Agocs et al. "Efficient method for solving highly oscillatory ordinary differential equations with applications to physical systems". In: *Phys. Rev. Research* 2 (1 Jan. 2020), p. 013030.

Error estimates



Gauss–Lobatto integration

Gaussian quadrature

$$\int_a^b w(x)f(x)dx \simeq \sum_{i=1}^{n} w_i f(x_i)$$

Gauss – Lobatto quadrature

$$\begin{array}{ll} interval(a,b): & [-1,\ 1] \\ w(x): & 1 \\ polynomials: & P'_{n-1}(x) \end{array}$$

Gauss – Lobatto quadrature

$$\int_{-1}^{1} f(x)dx \simeq \frac{2}{n(n-1)}(f(-1)+f(1)) + \sum_{i=2}^{n-1} w_i f(x_i)$$

$$\int_a^b f(x)dx \simeq \frac{b-a}{2}\left[\frac{2(f(a)+f(b))}{n(n-1)} + \sum_{i=1}^{n} w_i f\left(\frac{b-a}{2}x_i + \frac{b+a}{2}\right)\right]$$

Extended WKB

$$\ddot{x} + 2\gamma\dot{x} + T^2\omega^2 x = 0. \tag{1}$$

$$x(t) \sim \exp\left(T\sum_{n=0}^{\infty} S_n(t)T^{-n}\right). \tag{2}$$

$$\dot{S}_0(t) = \pm i\omega, \tag{3}$$

$$\dot{S}_i(t) = -\frac{1}{2S_0'}\left(\ddot{S}_{i-1} + 2\gamma\dot{S}_{i-1} + \sum_{j=1}^{i-1}\dot{S}_j\dot{S}_{i-j}\right). \tag{4}$$