oscode: fast solutions of oscillatory ODEs in cosmology

Fruzsina J. Agocs * [1,2]

[1]e-mail: `fa325@cam.ac.uk` [2]Kavli Institute for Cosmology, Cambridge [3]Astrophysics Group, Cavendish Laboratory, Cambridge

`oscode` is a Python/C++ package for the fast solution of oscillatory ordinary differential equations. It handles equations of the form $\ddot{x}(t) + 2\gamma(t)\dot{x}(t) + \omega^2(t)x(t) = 0$. The time-dependence of the frequency ω and the damping term γ can be explicit or implicit; below are examples of both. The algorithm is significantly more efficient than conventional (Runge–Kutta-based) solvers found in numerical libraries, thanks to reduced number of steps needed to traverse highly oscillatory regions.

Airy equation

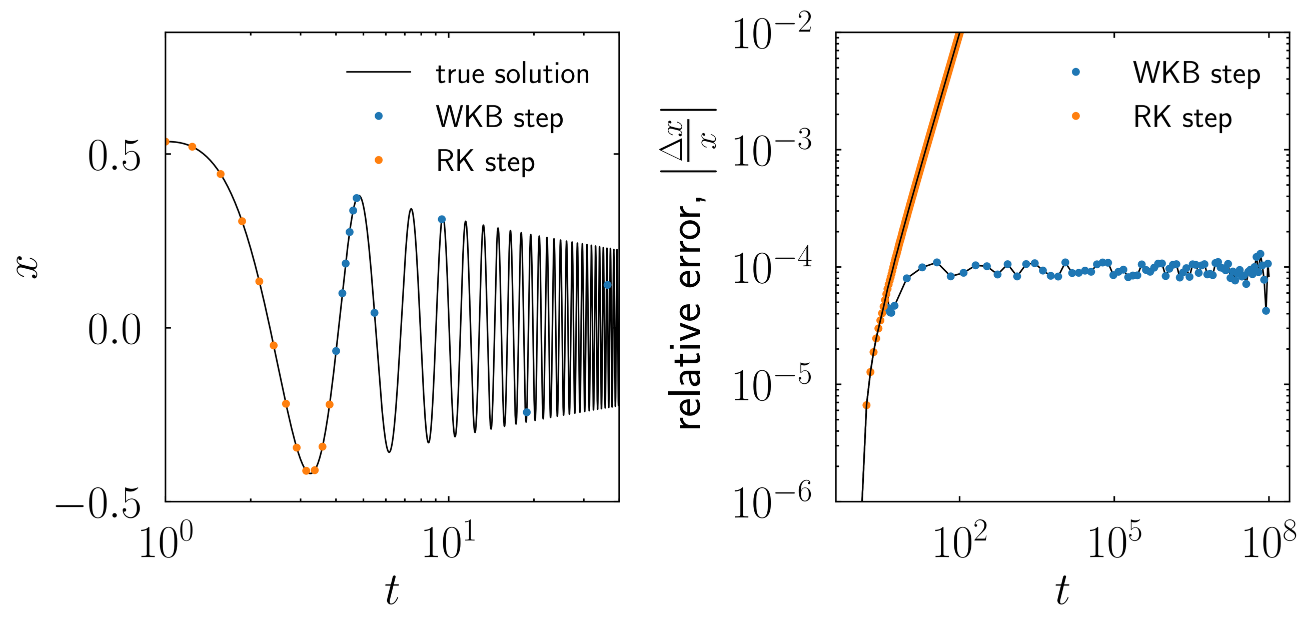

Figure 1: Steps and accuracy of `oscode` solving $\ddot{x} + tx = 0$.

- **Left:** different coloured dots show the internal steps `oscode` takes in `RK' and `WKB' mode, the continuous line being the exact solution
- **Right:** relative accuracy of `oscode`'s approximate solution
- Compare to scenario where `oscode` only takes `RK' steps

Burst equation

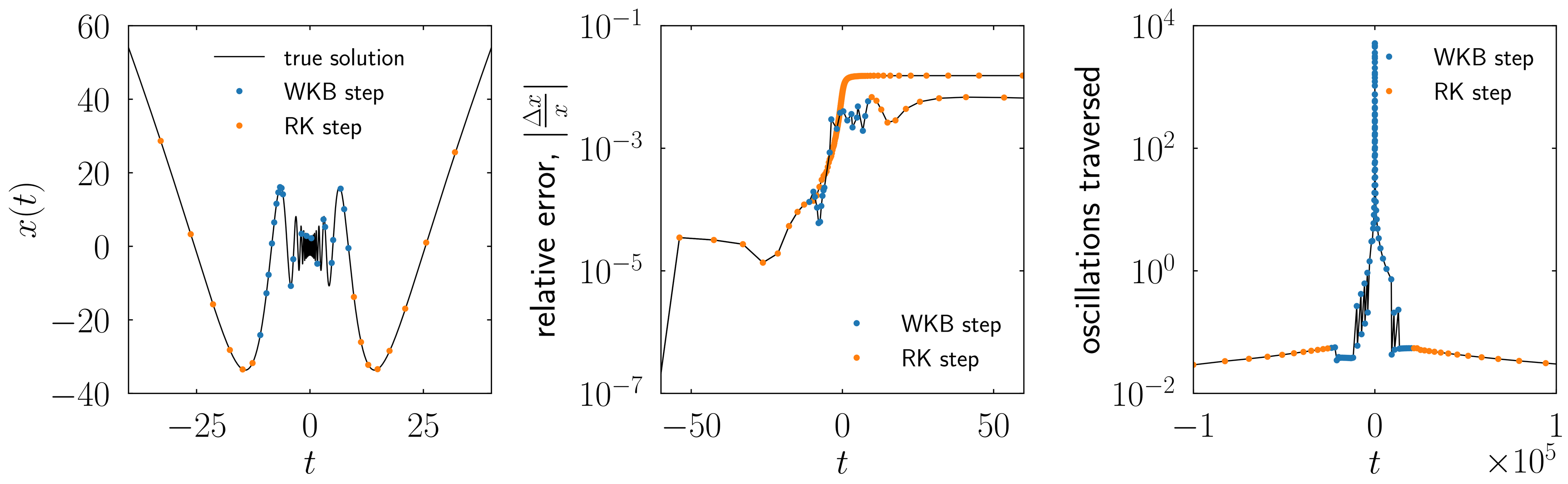

Figure 2: Solving $\ddot{x} + \frac{n^2-1}{(1+t^2)^2}x = 0$, an equation exhibiting a burst of oscillations.

- Equation has a parameter n, $n \sim$ no. of oscillations in total
- **Left:** `oscode`'s internal steps for $n = 40$
- **Middle:** Relative error in `oscode`'s solution, and error in `pure RK' solution for comparison
- **Right:** number of oscillations crossed per step for $n = 10^5$

Schrödinger equation

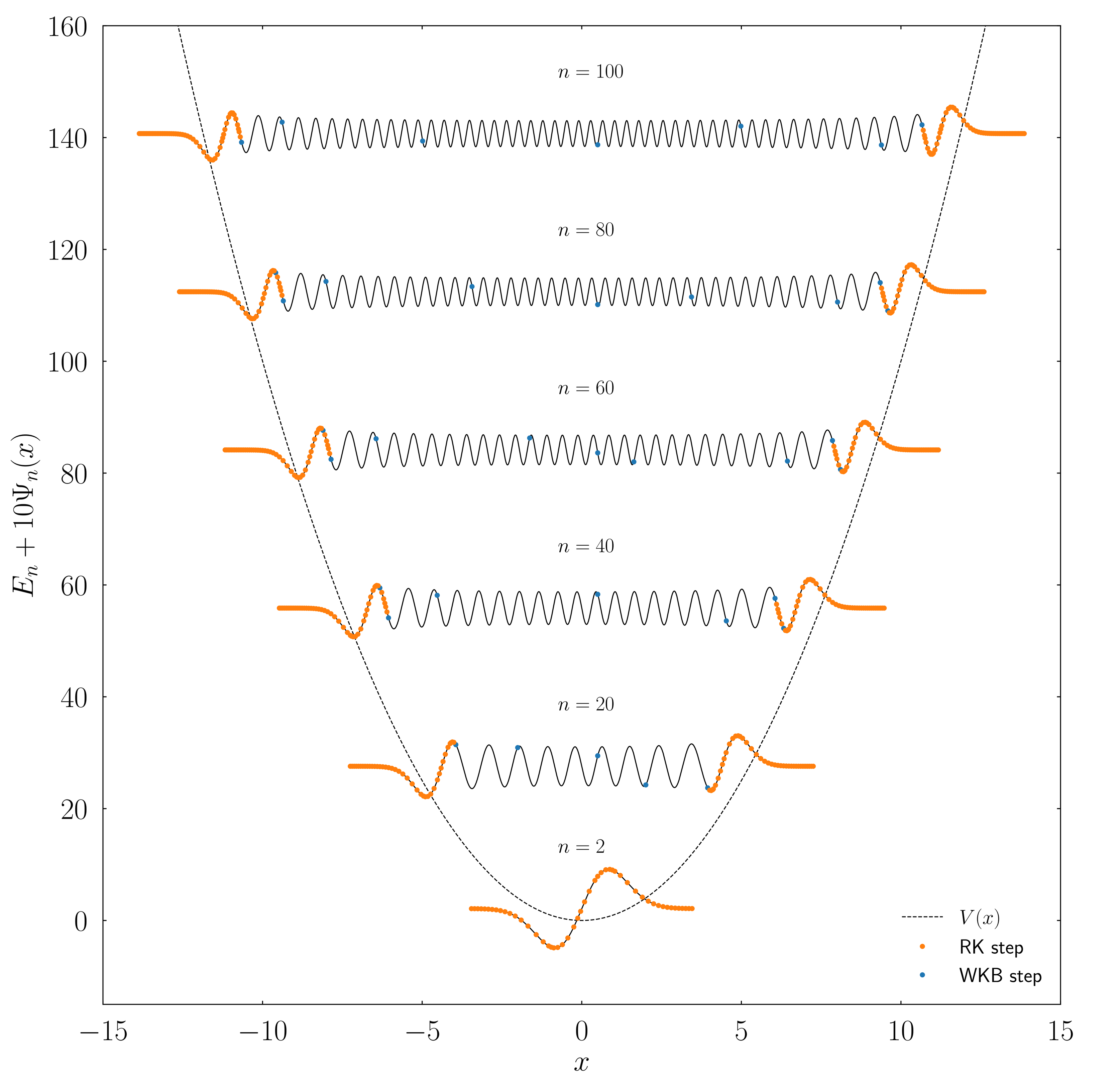

Figure 3: Eigenstates of the Schrödinger equation ($\Psi'' + 2m(E - V(x))\Psi = 0$) in a harmonic potential well.

- Above is an example where the eigenenergies are known, but can compute them numerically:
- Start integration from well outside potential well on either side and meet at $x = 0.5$
- Compute Ψ'/Ψ at the meeting point for both solutions; the difference is minimal if the energy is an eigenvalue
- Calculated energy eigenvalues in a harmonic well with quartic anharmonicity up to the 10000$^{\text{th}}$ energy level to 1 in 10^7 accuracy

Algorithm / Glossary

- **RK (Runge–Kutta) step:** approximates the solution at a later time $t + h$ as a Taylor expansion around t.
- **WKB (Wentzel–Kramers–Brillouin) step:** uses an analytic approximation which gets better the slower the frequency of oscillations change, allowing `oscode` to traverse highly oscillatory regions in few steps.
- `oscode` switches between RK/WKB mode, choosing whichever allows for the largest step within a given error tolerance.
- The stepsize is updated to ensure the local error stays within the given tolerance.
- Derivatives of the frequency and damping term are required for both the RK and WKB steps, these are computed numerically as finite differences.
- WKB steps also require integrals, which are computed using Gauss–Lobatto integration.
- The RK steps and finite difference formulae were designed to minimise function evaluations.

Primordial power spectra

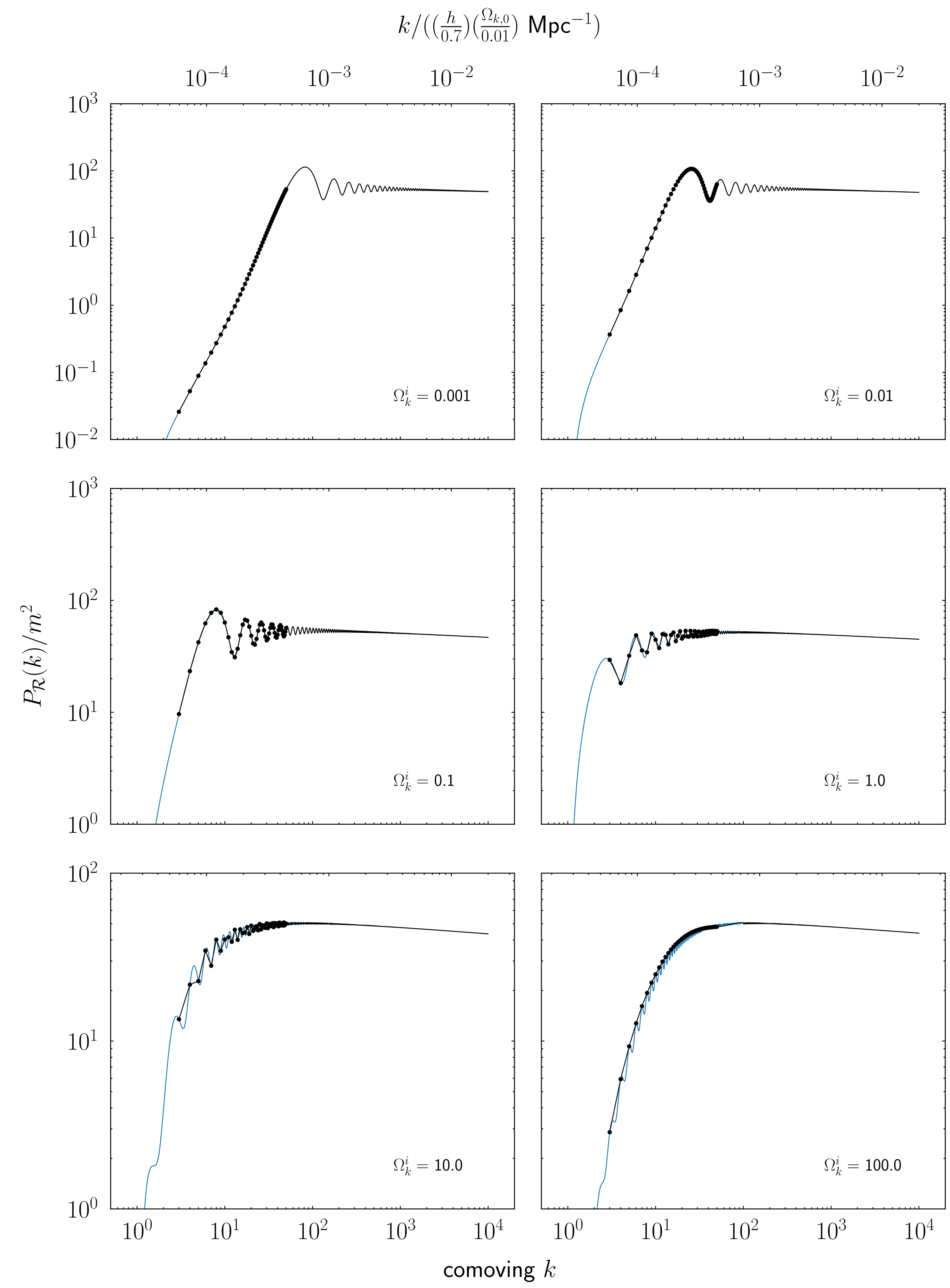

Figure 4: Scalar primordial power spectra in closed universes with varying start-of-inflation curvature Ω_k^i.

- Solve the Mukhanov–Sasaki (MS) equation which governs the time-evolution of spacetime-curvature perturbations in the early (pre-inflationary) Universe
- perturbations characterised by wavenumber k, need to solve for each k separately to get *primordial power spectrum*
- MS equation has a freqency roughly $\propto k$, solution becomes extremely slow with off-the-shelf integrators at large k
- Above examples are primordial power spectra of closed universe models, for which `oscode` gives a speedup of $\geq \mathcal{O}(10^3)$, depending on parameters
- This allows for faster inference of parameters and exploration of more models

For paper, code and an animated version of Fig. 4, take a photo of this QR code or visit `https://github.com/fruzsinaagocs/oscode`