>>> Efficient numerical solutions for oscillatory differential equations

Name: Fruzsina Agocs (KICC and Cavendish AP group) ¶

Date: December 22, 2019

¶fa325@cam.ac.uk

\>>> Overview

1. Introduction
2. Runge-Kutta
3. Wentzel-Kramers-Brillouin
4. RKWKB
5. Challenges

>>> The problem and why it matters

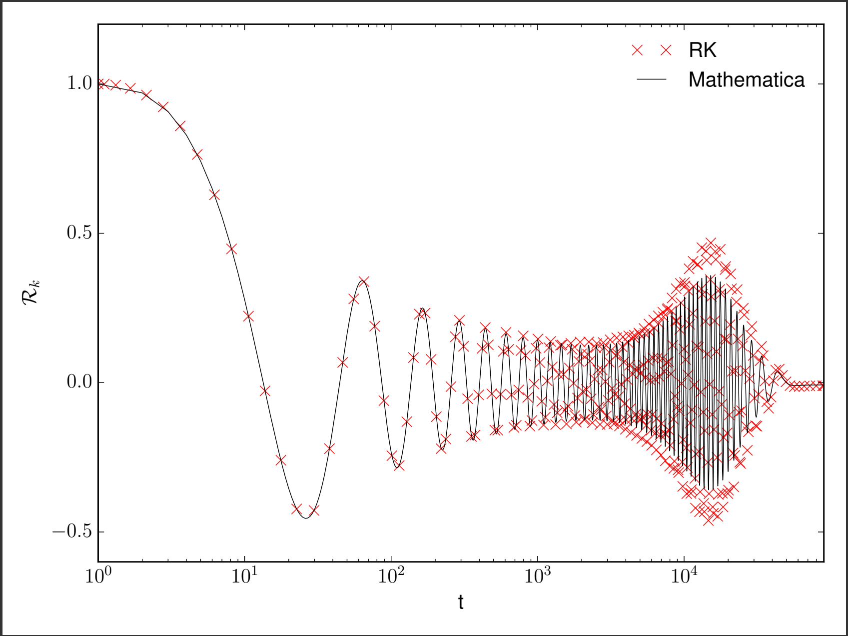
ODE-s with oscillatory solutions abound in physics:

* Quantum mechanics
* Plasma physics, MHD
* Cosmology: Boltzmann equations, Mukhanov-Sasaki equation

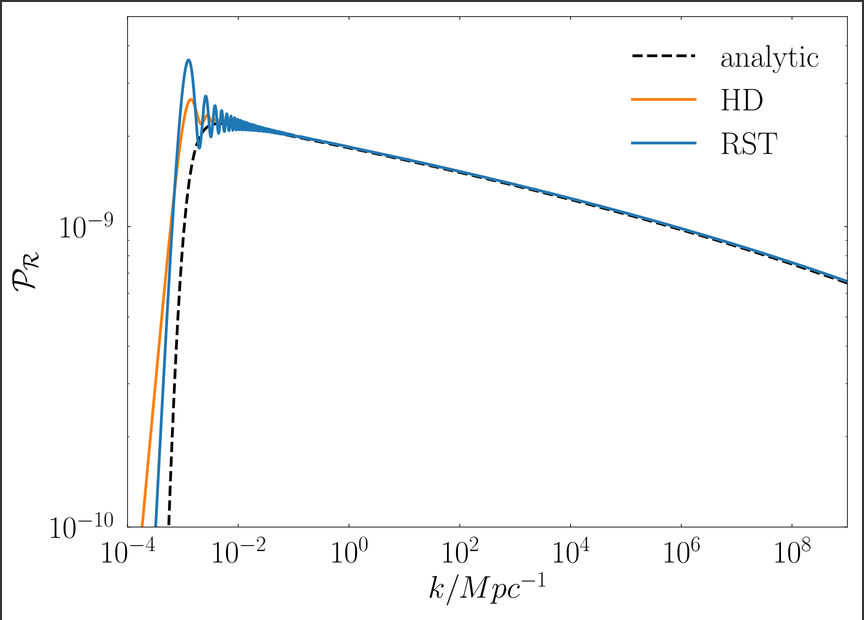
$$\ddot{x} + 2\gamma(t)\dot{x} + \omega^2(t)x = 0 \tag{1}$$

$$\begin{gathered}\ddot{x} + 2\gamma(\vec{y})x + \omega^2(\vec{y})x = 0 \\ \dot{\vec{y}} = \vec{F}(\vec{y})\end{gathered} \tag{2}$$

>>> An example



\>>> An example



analytic
HD
RST
10^{-9}
10^{-10}
$\mathcal{P}_{\mathcal{R}}$
10^{-4}
10^{-2}
10^{0}
10^{2}
10^{4}
10^{6}
10^{8}
k/Mpc^{-1}

\>>> The problem and why it matters

* Conventional methods are not efficient at solving these
* Fast computation would allow one to
  * evaluate likelihoods faster,
  * explore a larger parameter space,
  * not use approximations

>>> Why Runge-Kutta is so successful

At the heart of Runge-Kutta methods is a simple idea:

$$y_{i+1} = y_i + hF_i + \frac{h^2}{2}\left(\frac{dF}{dt}\right)_{t_i} + \mathcal{O}(h^3)$$

simulated by

$$y_{i+1} = y_i + b_1 hF_i + b_2 hF(t_i + c_1 h, y_i + a_1 hF_i).$$

* Taylor expand around t_i, y_i,
* match coefficients of h.
* Pair with *adaptively updating* h[1]

[1] more about this later.

>>> The Wentzel-Kramers-Brillouin approximation

* So why not use Runge-Kutta?
  * polynomials might not be the most suitable trial solutions
* Try instead:

$$f(t) \sim A(t)e^{i\phi(t)}$$

for a *slowly changing* frequency:

$$f_{\pm}(t) \sim \frac{1}{\sqrt{\omega(t)}} e^{\pm i \int^t \omega(\tau) d\tau + \dots}$$

>>> Merging RK and WKB: the RKWKB method

* Embed the WKB approx. in a Runge-Kutta stepping procedure
* At each step, forecast solution:

$$x(t+h) = A_+ f_+(t) + A_- f_-(t)$$

$$\dot{x}(t+h) = B_+ \dot{f}_+(t) + B_- \dot{f}_-(t)$$

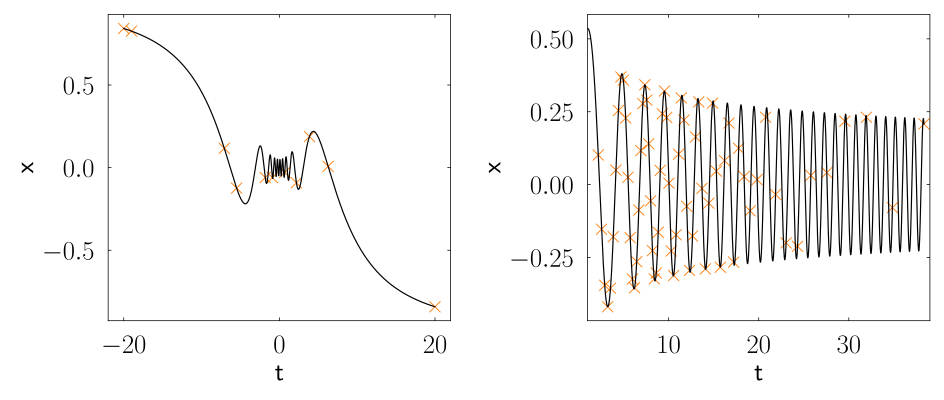
$$A_\pm, B_\pm \left(\{f_\pm, \dot{f}_\pm, \ddot{f}_\pm, x, \dot{x}\}_t \right)$$

* Approximate the error per step as Δ, then

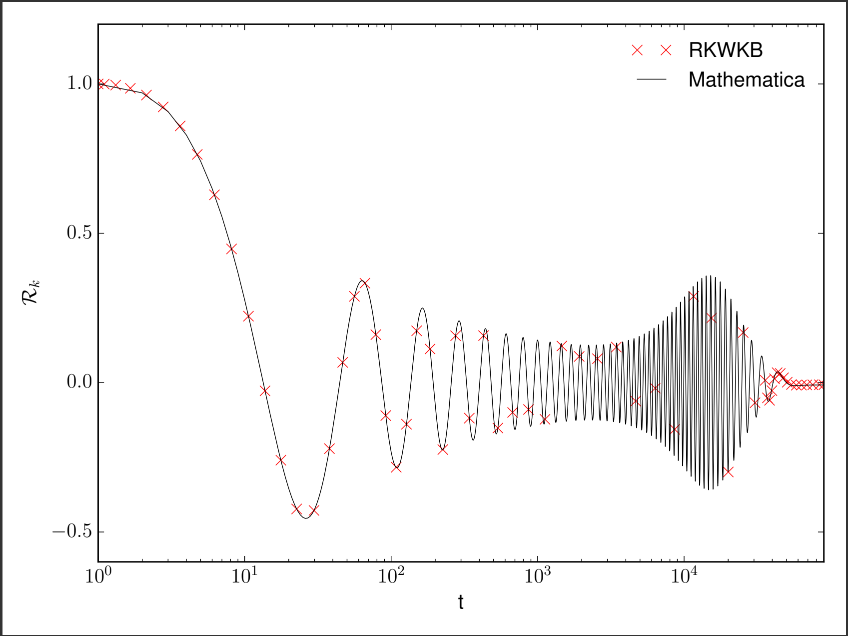
$$h = h \cdot \left(\frac{\Delta}{\Delta^*} \right)^{-1/N},$$

  if one expects $\Delta \sim h^N$ and Δ^* is a threshold.
* Switch between RK/WKB.

\>>> Examples



>>> Examples



>>> Challenges

* Higher order WKB approximations require $\dot{\omega}, \ddot{\omega}, \ldots$
* These are not always available as a fn of t:

$$\dot{\omega} = \nabla_{\vec{y}}\omega \cdot \dot{\vec{y}} = \nabla_{\vec{y}}\omega \cdot F(\vec{y})$$

* Can supply gradients, Hessians by hand -> cumbersome
* automatic differentiation -> slow?
* numerical differentiation

>>> Thank you!