Chirps and waves: adaptive high-order methods for oscillatory ODEs and PDEs

Fruzsina J Agocs

CU Boulder, Computer Science

CU Boulder Computational Tools group introduction

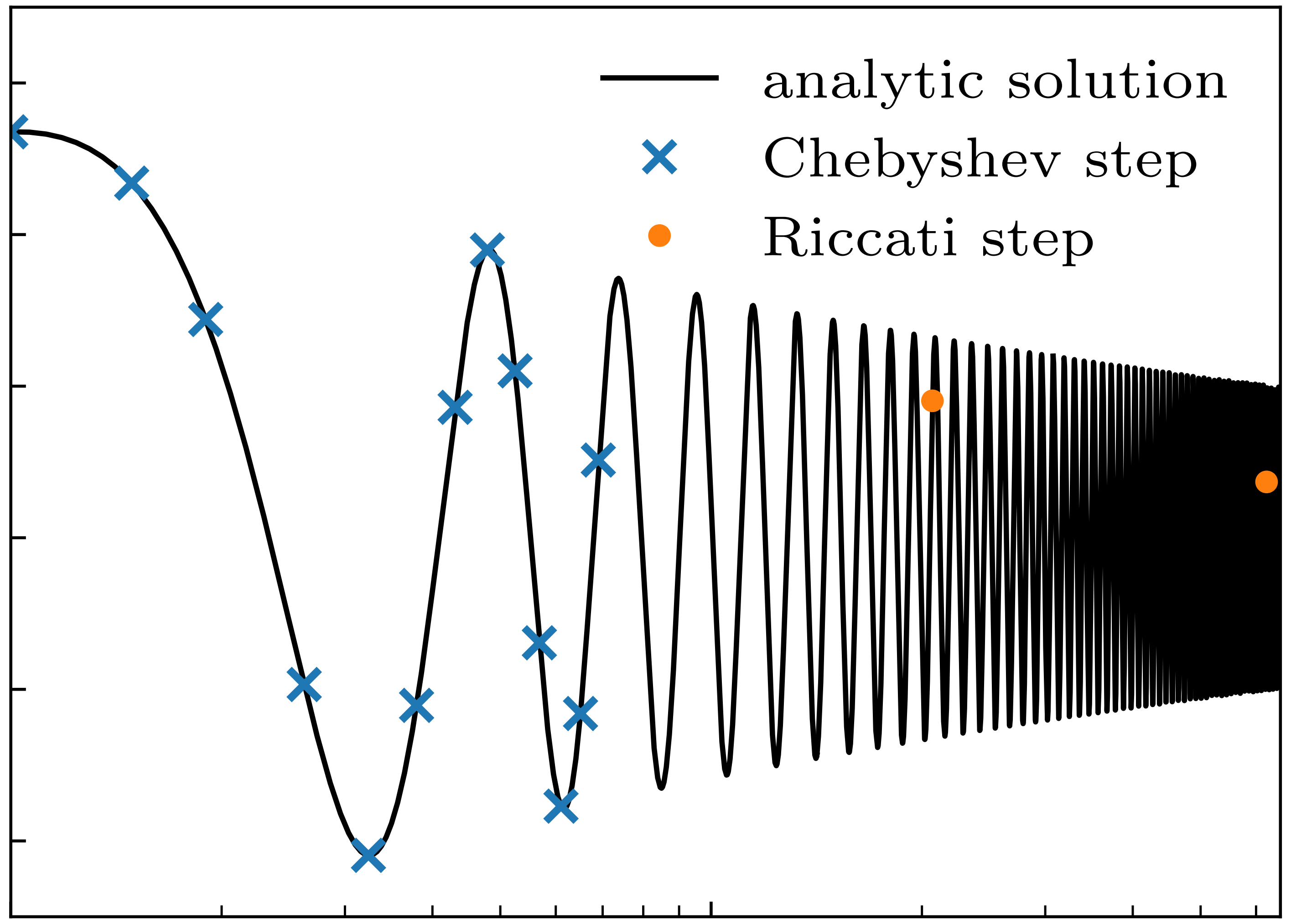
Research profile/principles

- I build numerical methods for physics/engineering applications
- Numerical analysis $\times$ scientific computing $\times$ computational physics
- Adaptive, efficient, high-order accurate methods for **ODEs, PDEs**
  - Stay efficient when high accuracy is demanded
  - Require little user input
  - Open-source software implementation
- Insight into current problems, computational bottlenecks in **computational astrophysics**

Contents — my contributions and interests

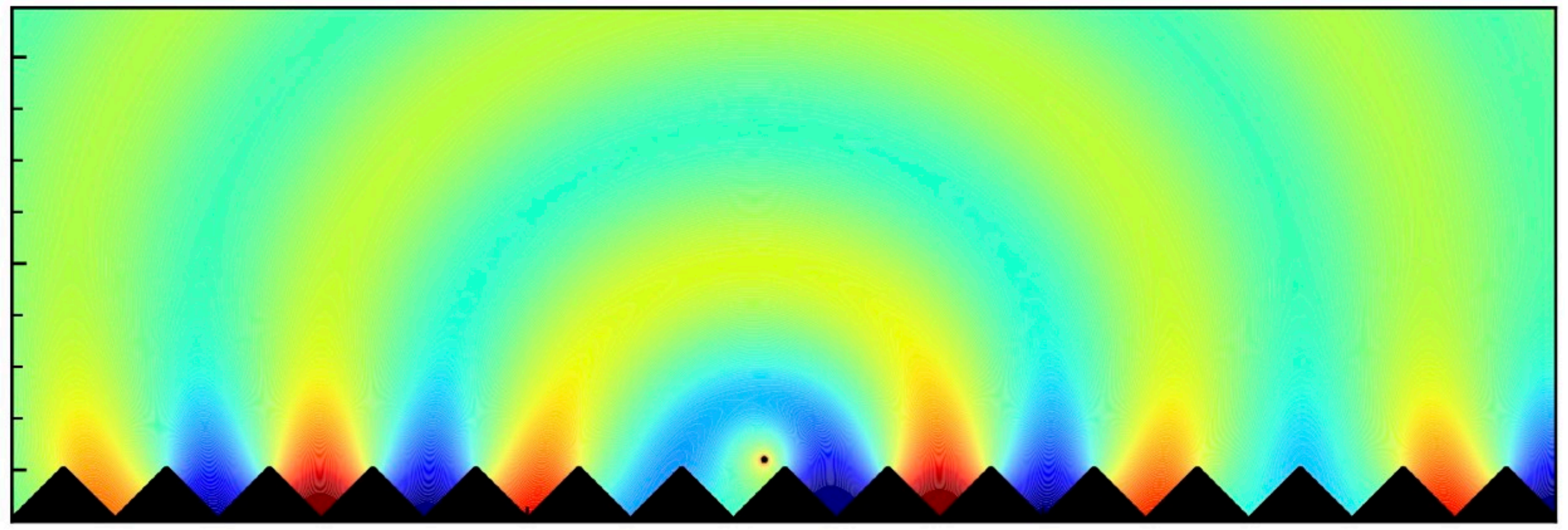
ODEs

- Fast, high-order accurate, adaptive solvers for ODEs with oscillatory solutions, software
- *Motivation:* Special function evaluation, repeated oscillatory ODE solves in optimization and inference in physics



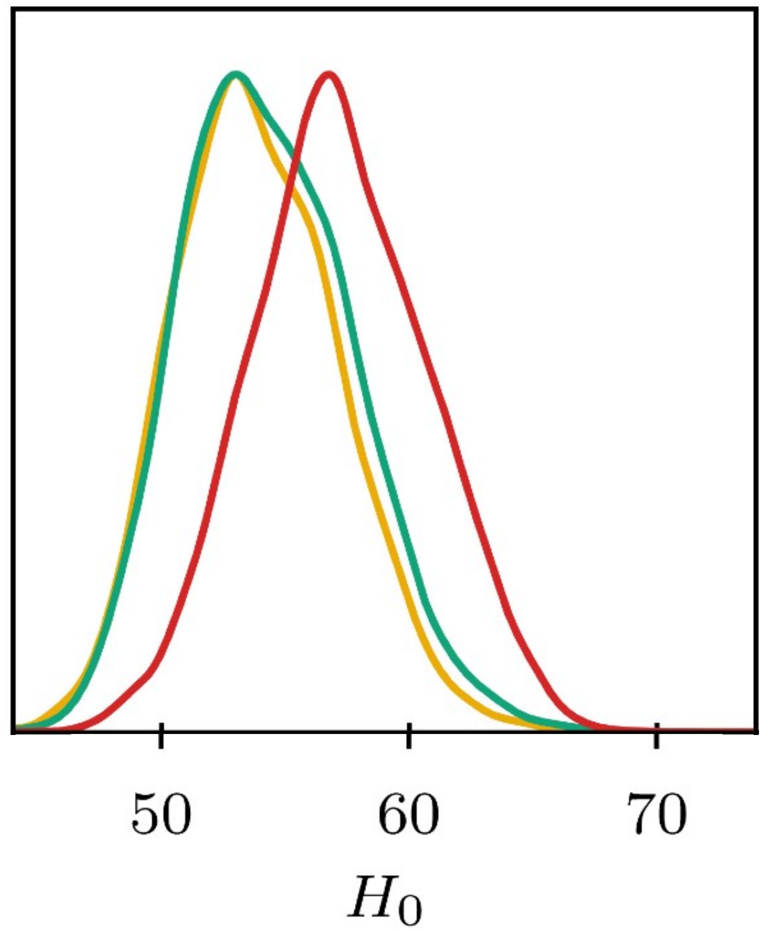
PDEs

- Fast, high-order boundary integral equation methods for PDEs in complex geometries
- scattering from periodic surfaces with corners; periodic and nonperiodic sources
- *Motivation:* waveguides, acoustic/seismic filtering, remote sensing, topological insulators, fault detection, material design



Computational cosmology

- Theoretical foundations (QFT)
- Fast forward modeling
- Bayesian inference from particle physics x cosmology data (GAMBIT)
- *Motivation:* What's dark matter? What was physics like in the early universe? What's beyond the SM?



ODEs

Frequency-independent solver for oscillatory ODEs

The problem and why it's hard

- Build an efficient solver for second-order, linear, homogeneous ODEs of the form

$$u''(t) + 2\gamma(t)u'(t) + \omega^2(t)u(t) = 0, \quad t \in [t_0, t_1],$$
$$u(t_0) = u_0, \quad u'(t_0) = u'_0\,.$$

From now on, set $\gamma(t) = 0$ for simplicity.

- Assumptions: $\omega(t)$, $\gamma(t)$ real-valued, $\omega(t) \geq 0$, $\omega(t)$ is **large and slowly-varying** for some of $[t_0, t_1]$.
- Conventional* methods need $\mathcal{O}(1/\omega)$ discretization length $\rightarrow$ $\mathcal{O}(\omega)$ runtime, **prohibitively slow** at large ω!
- But small $\omega(t)$, large $\gamma(t)$ results in nonoscillatory behavior; solver needs to handle this

Goals and motivation

- Build **4-6th- and high-order, adaptive, efficient** solvers, $\mathcal{O}(1)$ (**frequency-independent**) runtime
- User-friendly, open-source software
- Ubiquitous: **cosmology**, quantum mech, **special functions**, electric circuits, …
- Often part of Bayesian inference or optimization loop $\rightarrow$ 10^6–10^9 solves needed

Previous work and my contributions

Existing specialized oscillatory solvers

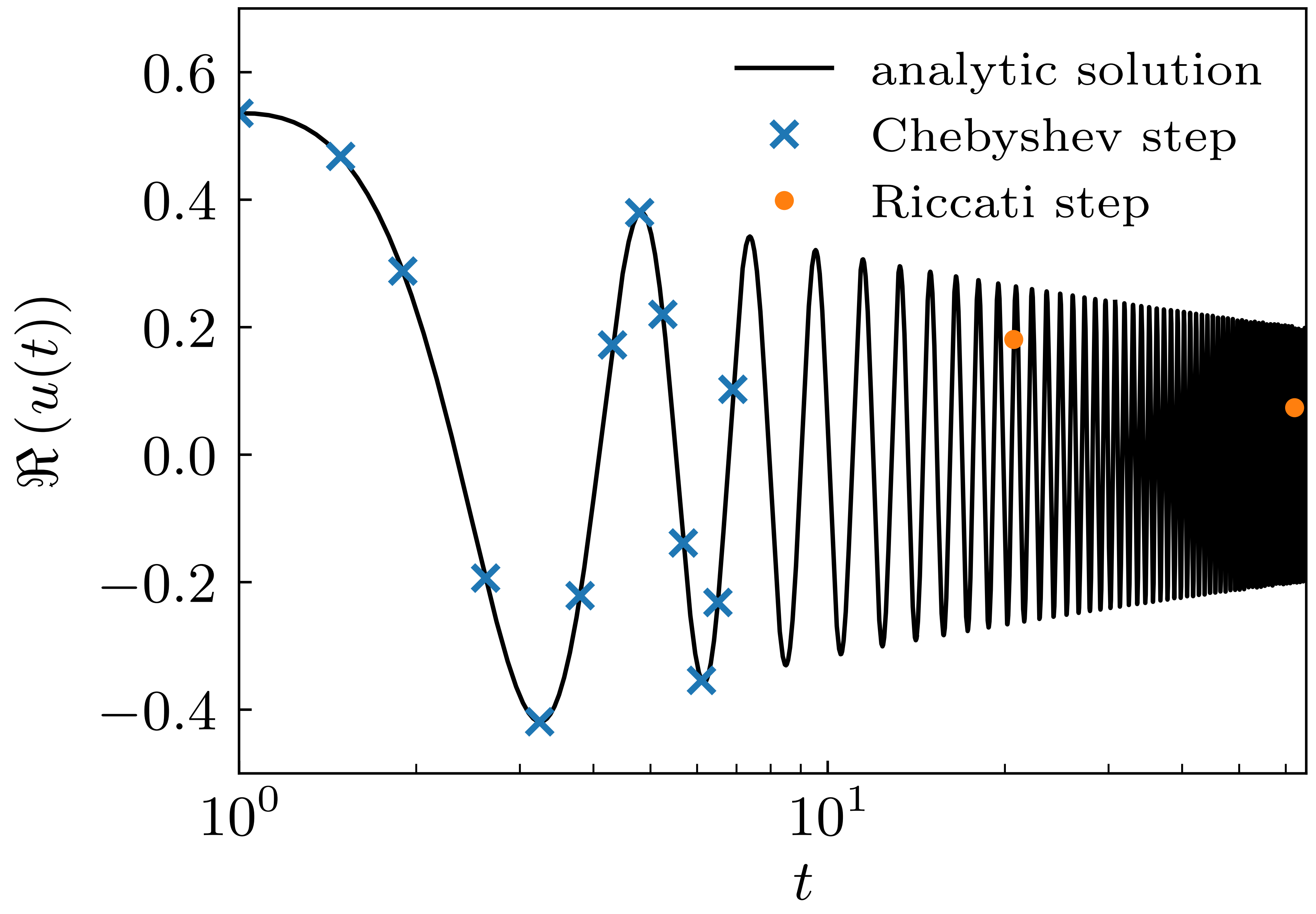
- None can deal with $\omega(t)$ changing significantly in magnitude
- Other high-order specialized oscillatory solvers exist, but only for $\omega(t) \gg 1$ (e.g. *Bremer, ACHA, 2018.*)
- Asymptotic expansions for oscillatory functions, applied analytically (“by hand”) and at low order
- **No software in large numerical libraries!**

My contributions

- 4-6th-order, adaptive method for 3-6 digits of accuracy, physics applications: *Agocs et al, Phys Rev Research, 2020.,* arXiv:1906.01421
  - Includes novel switching algorithm
- Arbitrarily high-order, adaptive method for higher accuracy: *Agocs & Barnett, SINUM, 2023.,* arXiv:2212.06924
  - Includes new, simple asymptotic expansion, and error bound
- Software: `oscode` (10.21105/joss.02830), `riccati` (10.21105/joss.05430)

Methods overview

- Time-stepping with **adaptive stepsize** $h(t)$, keep local error estimate below user-defined (relative) **tolerance** ε
- Right strategy: exploit known behavior of the solution $u(t)$ and always work in terms of a slowly-varying quantity
  - $\rightarrow$ larger timesteps, $\mathcal{O}(1)$ runtime
- Two different methods for $u(t)$ oscillatory or slowly-varying
  - $\omega \lessapprox 1$: spectral collocation on Chebyshev nodes / 4-5th order Runge—Kutta
  - $\omega \gg 1$: asymptotic expansion: **Riccati defect correction** or Wentzel—Kramers—Brillouin (WKB)
- Automatic **switching** between the two methods: choose whichever maximizes stepsize h while still keeping the error below ε



Asymptotic expansions: Riccati defect correction

- Work in terms of slowly-varying "phase function" $x(t)$, defined as $u(t) = e^{\int^t x(\sigma)\mathrm{d}\sigma}$
- $x(t)$ obeys **Riccati eq:** $0 = x' + x^2 + \omega^2$, **oscillatory!**
- Construct approximate, non oscillatory phase function $x(t)$ by functional iteration: $x_0(t), x_1(t), \ldots, x_j(t)$
- If $\omega \gg 1$, start from the initial approximation $x_0(t) = \pm i\omega(t)$ (exact if $\omega(t)$ const)
- Define **residual of the Riccati eq.**:

$$R[x](t) := R[x] = x' + x^2 + \omega^2, \text{ then}$$

$$R[x_j + \delta] = R[x_j] + \cancel{\delta'} + 2x_j\delta + \cancel{\mathcal{O}(\delta^2)}$$

- Seek a function $\delta(t)$ that gives $R[x_j + \delta] = 0$
- Linearize, then neglect δ'[1], get **defect correction scheme**:

$$x_{j+1}(t) = x_j(t) \underbrace{- \frac{R[x_j](t)}{2x_j(t)}}_{\delta(t)}, \quad \text{for all } t \in [t_i, t_{i+1}]\,.$$

[1]If δ is non oscillatory (which we *choose* it to be), then δ' is $\mathcal{O}(\omega)$ smaller than all other terms.

Asymptotic expansions: Riccati defect correction

- Work in terms of slowly-varying "phase function" $x(t)$, defined as $u(t) = e^{\int^t x(\sigma)\mathrm{d}\sigma}$
- $x(t)$ obeys **Riccati eq:** $0 = x' + x^2 + \omega^2$, **oscillatory!**
- Construct approximate, non oscillatory phase function $x(t)$ by functional iteration: $x_0(t), x_1(t), \ldots, x_j(t)$
- If $\omega \gg 1$, start from the initial approximation $x_0(t) = \pm i\omega(t)$ (exact if $\omega(t)$ const)
- Define **residual of the Riccati eq.**:

$$R[x](t) := R[x] = x' + x^2 + \omega^2, \text{ then}$$

$$0 = R[x_j] + 2x_j\delta$$

- Seek a function $\delta(t)$ that gives $R[x_j + \delta] = 0$
- Linearize, then neglect δ'[1], get **defect correction scheme**:

$$x_{j+1}(t) = x_j(t) \underbrace{- \frac{R[x_j](t)}{2x_j(t)}}_{\delta(t)}, \quad \text{for all } t \in [t_i, t_{i+1}]\,.$$

Summary:

Change variables from $u(t)$ to "phase" $x(t)$

Construct an (asymptotic) approximation for the phase

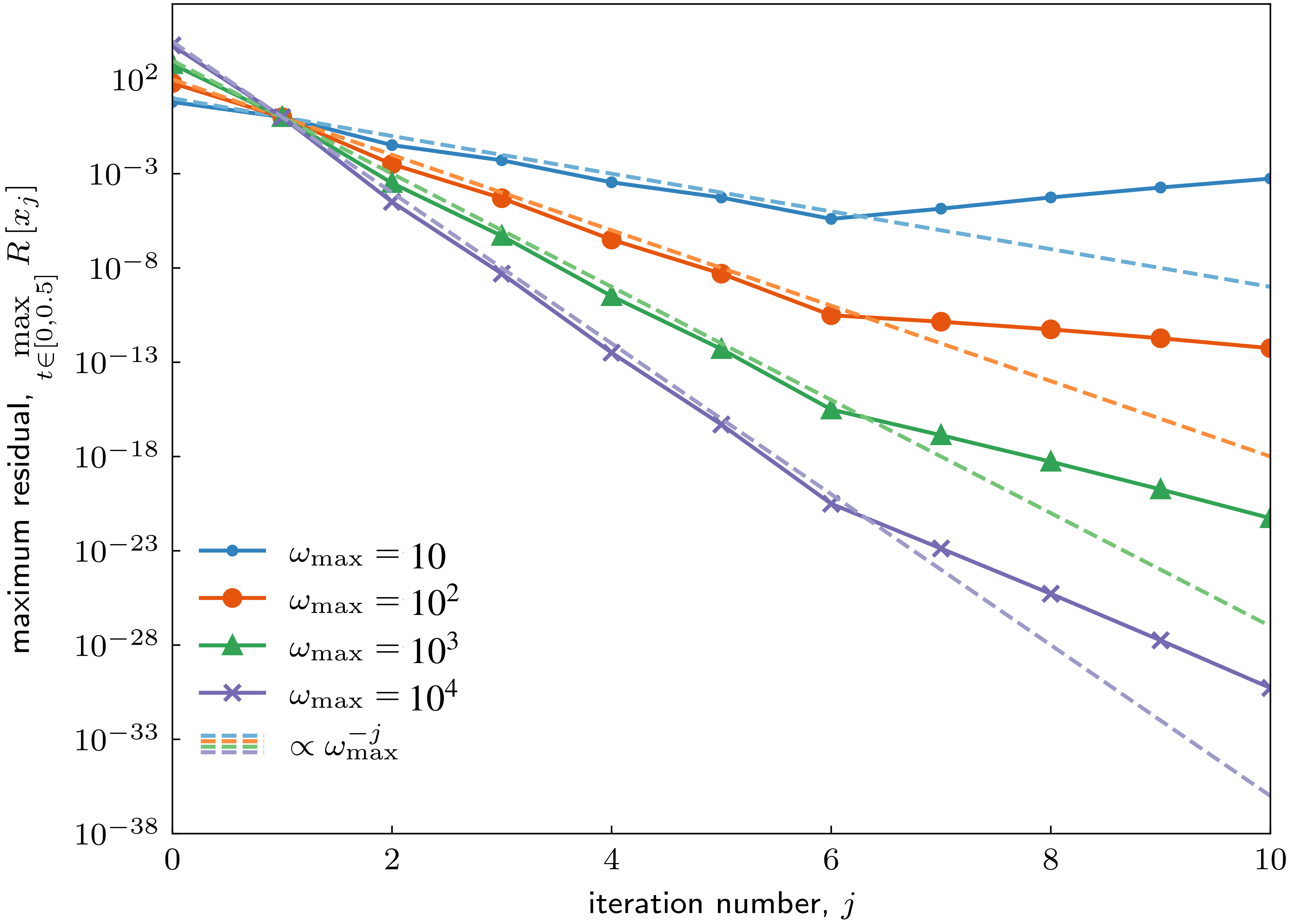
Approximation is especially good when ODE coefficients are smooth and large

This results in large stepsizes

[1] If δ is non oscillatory (which we *choose* it to be), then δ' is $\mathcal{O}(\omega)$ smaller than all other terms.

Asymptotic expansions: Riccati defect correction

$$u'' + \frac{\omega_{\max}^2}{1+t^2} u = 0$$



- Let ω be analytic in the closed ball $B_\rho := \{z \in \mathbb{C} : |z - t| \leq \rho\}$ centered on a given t.
- Then for $j = 1, 2, \ldots, k$,

$$R_j(t) \leq Ar^j, \text{ with}$$

$$r(|\omega'|_{B_\rho}, |\omega|_{B_\rho}, k)\,.$$

- If $|\omega'|/|\omega|$ is small and $|\omega|$ is large in B_ρ, then geometric reduction of residual up to $j \leq k$ iterations.

Asymptotic expansions: Riccati defect correction

- Once we have x_j, transform back:

$$u(t) = e^{\int^t x_j(\sigma)\mathrm{d}\sigma}$$

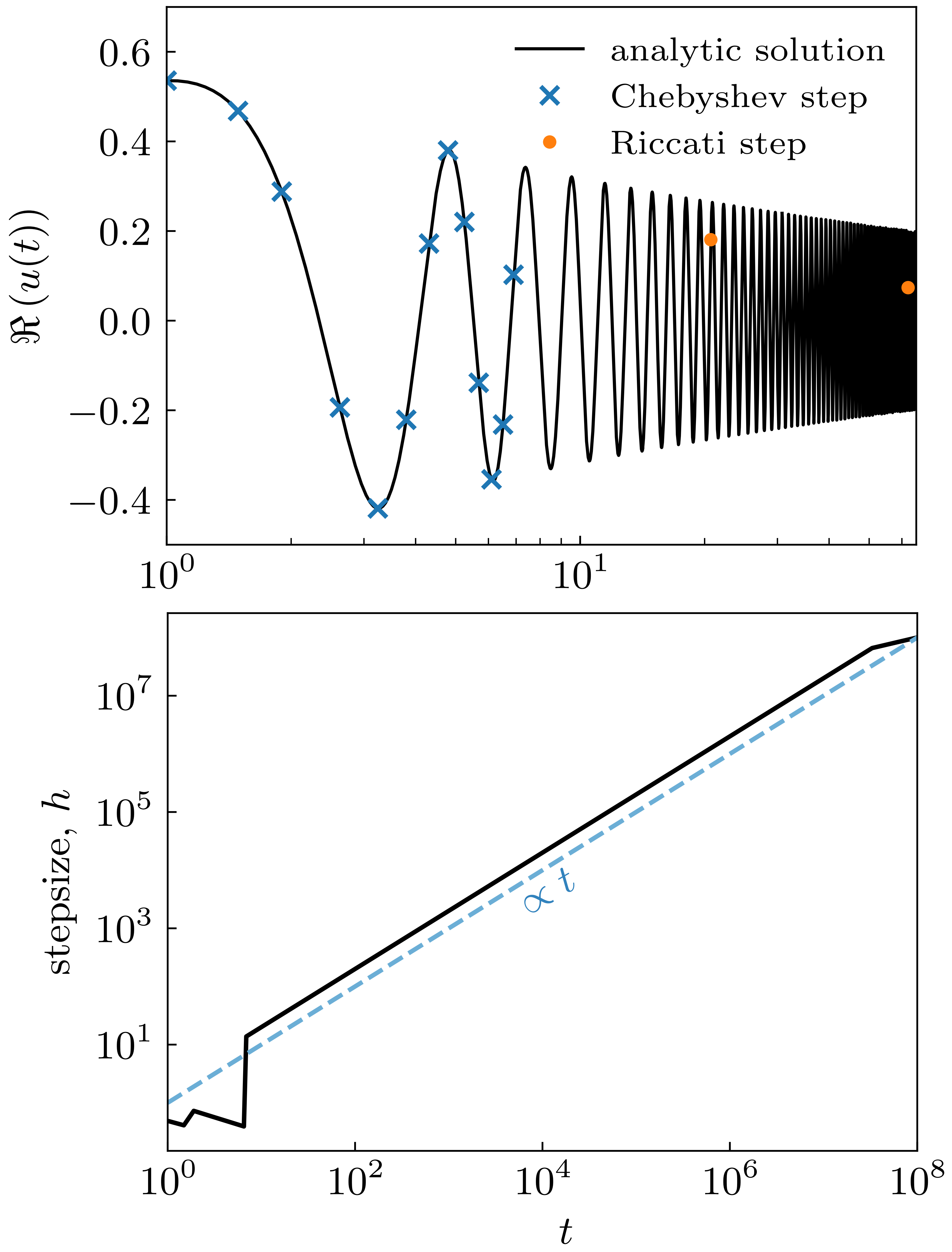
- Two solutions for x_j: $x_{j\pm}$ (starting from $\pm i\omega$) give linearly independent solutions for u; $u_\pm$:
- Linearly combine to match initial conditions at the start of each timestep:

$$u(t_{i+1}) = Au_+ + Bu_-, \quad u'(t_{i+1}) = Au'_+ + Bu'_-$$

- Error estimate is via residual $R[x_j]$. Fix stepsize, iterate over j.
- Derivatives and integral via spectral differentiation / integration matrix with $n = 16$ nodes.
- Right: to 12 digits, solve

$$u'' + tu = 0, \quad t \in [1,10^8]$$

$$u(t) = \mathrm{Ai}(-t) + i\mathrm{Bi}(-t)$$



Asymptotic expansions: Riccati defect correction

- Once we have x_j, transform back:

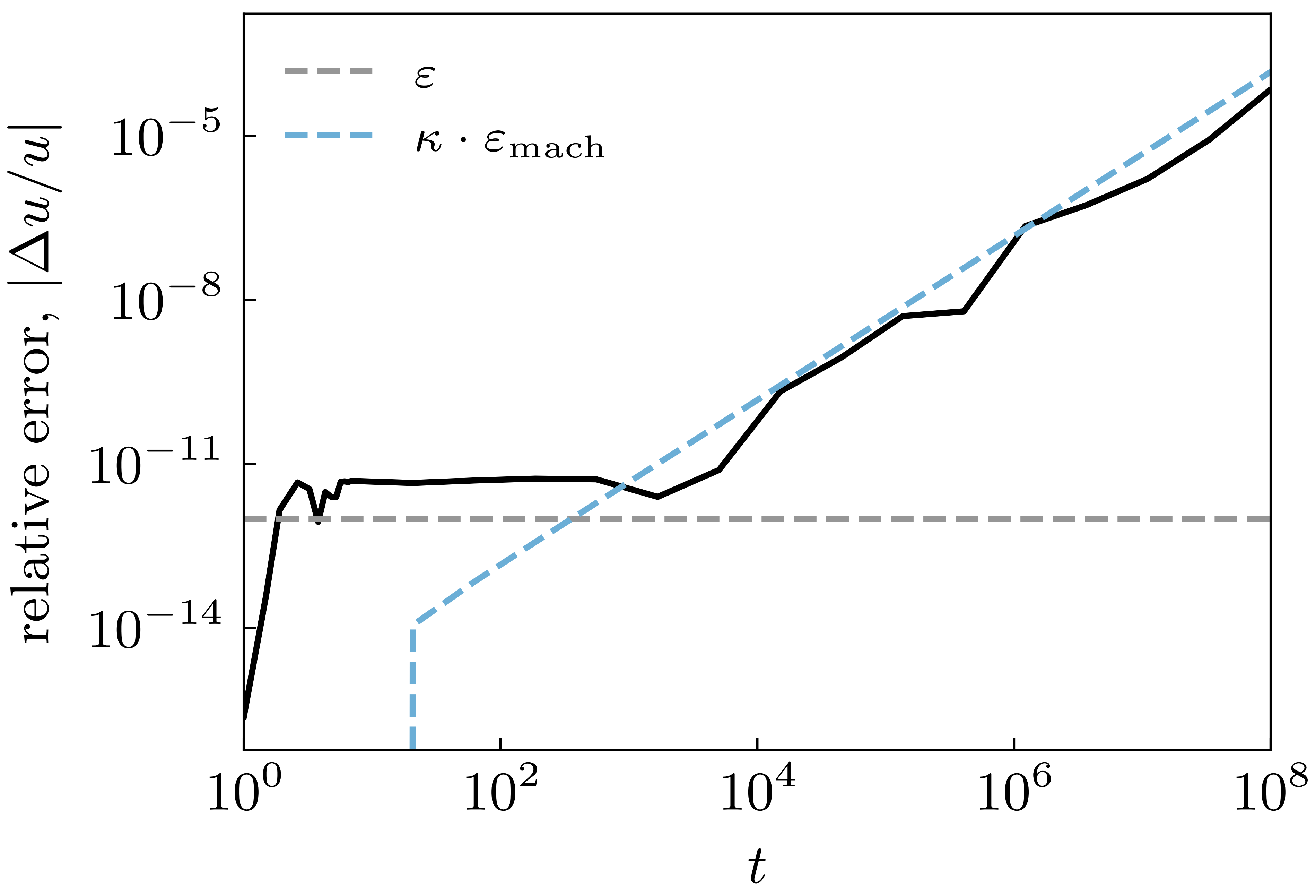
$$u(t) = e^{\int^t x_j(\sigma)\mathrm{d}\sigma}$$

- Two solutions for x_j: $x_{j\pm}$ (starting from $\pm i\omega$) give linearly independent solutions for u; $u_\pm$:
- Linearly combine to match initial conditions at the start of each timestep:

$$u(t_{i+1}) = Au_+ + Bu_-, \quad u'(t_{i+1}) = Au'_+ + Bu'_-$$

- Error estimate is via residual $R[x_j]$. Fix stepsize, iterate over j.
- Derivatives and integral via spectral differentiation / integration matrix with $n = 16$ nodes.
- Right: to 12 digits, solve

$$u'' + tu = 0, \quad t \in [1, 10^8]$$
$$u(t) = \mathrm{Ai}(-t) + i\mathrm{Bi}(-t)$$



Asymptotic expansions: Riccati defect correction

- Once we have x_j, transform back:

$$u(t) = e^{\int^t x_j(\sigma)\mathrm{d}\sigma}$$

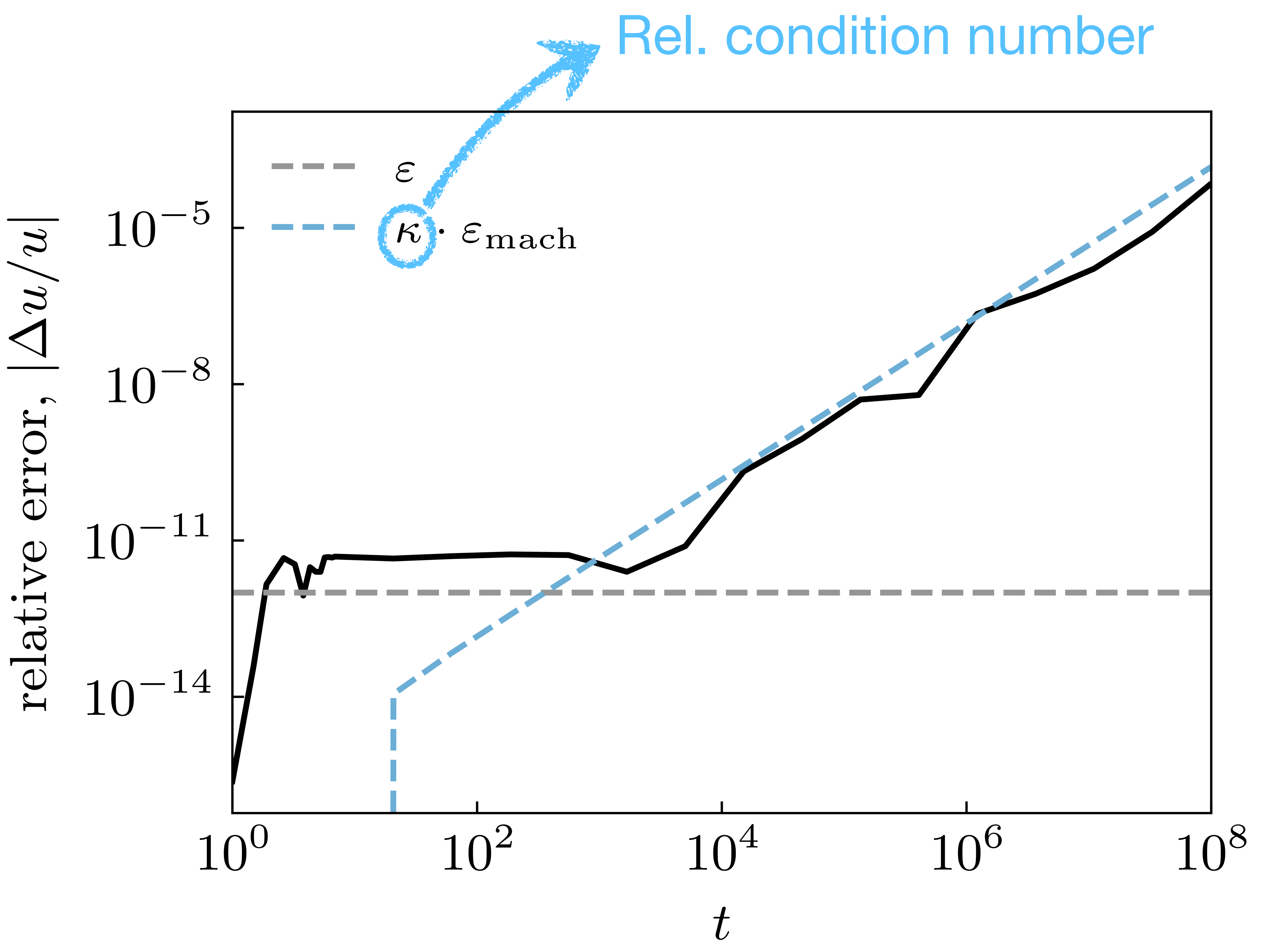
- Two solutions for x_j: $x_{j\pm}$ (starting from $\pm i\omega$) give linearly independent solutions for u; $u_\pm$:
- Linearly combine to match initial conditions at the start of each timestep:

$$u(t_{i+1}) = Au_+ + Bu_-, \quad u'(t_{i+1}) = Au'_+ + Bu'_-$$

- Error estimate is via residual $R[x_j]$. Fix stepsize, iterate over j.
- Derivatives and integral via spectral differentiation / integration matrix with $n = 16$ nodes.
- Right: to 12 digits, solve

$$u'' + tu = 0, \quad t \in [1,10^8]$$

$$u(t) = \mathrm{Ai}(-t) + i\mathrm{Bi}(-t)$$

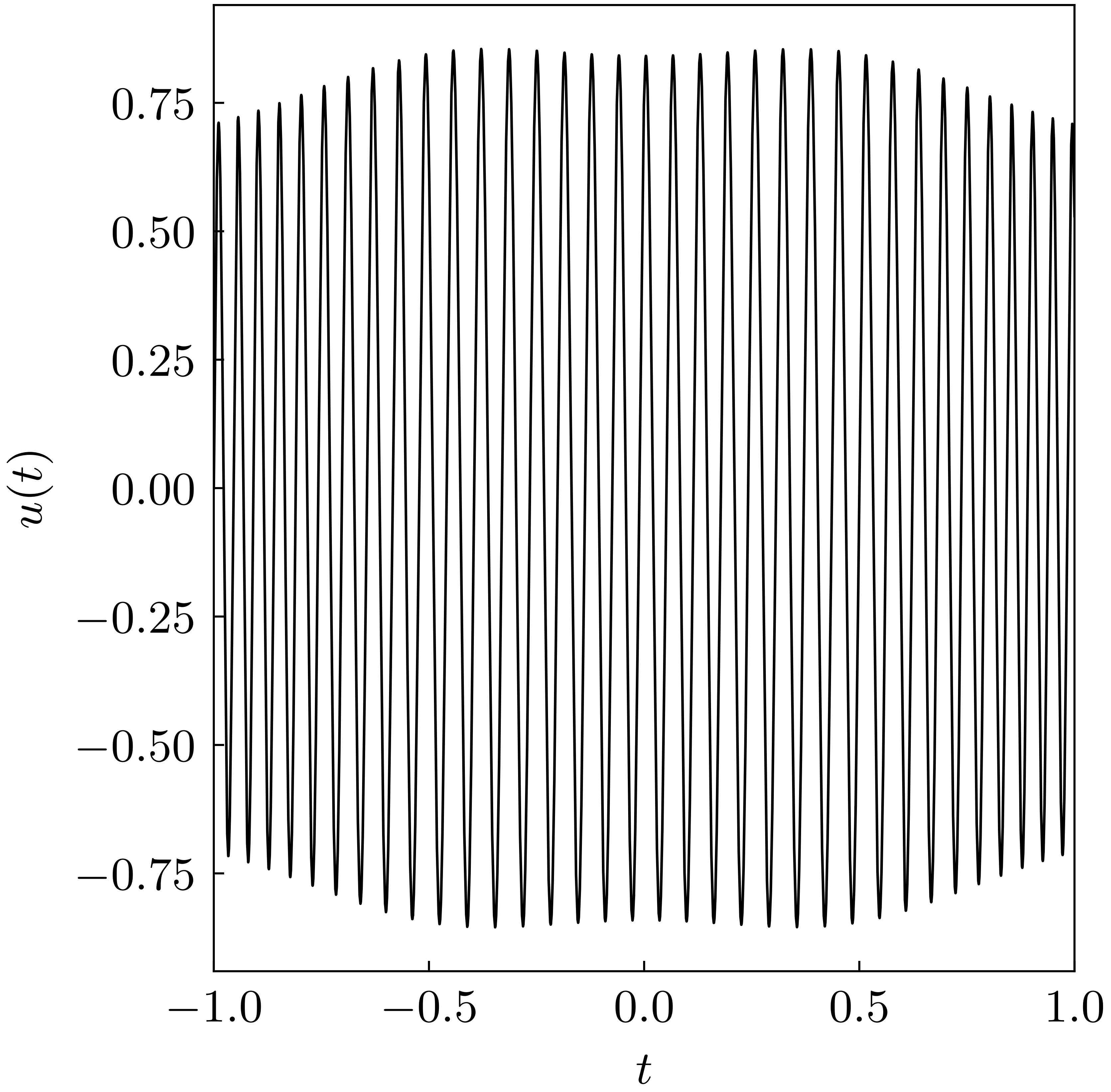


Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

$$u''(t) + \lambda^2 q(t)u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2 \cos(3t)$$

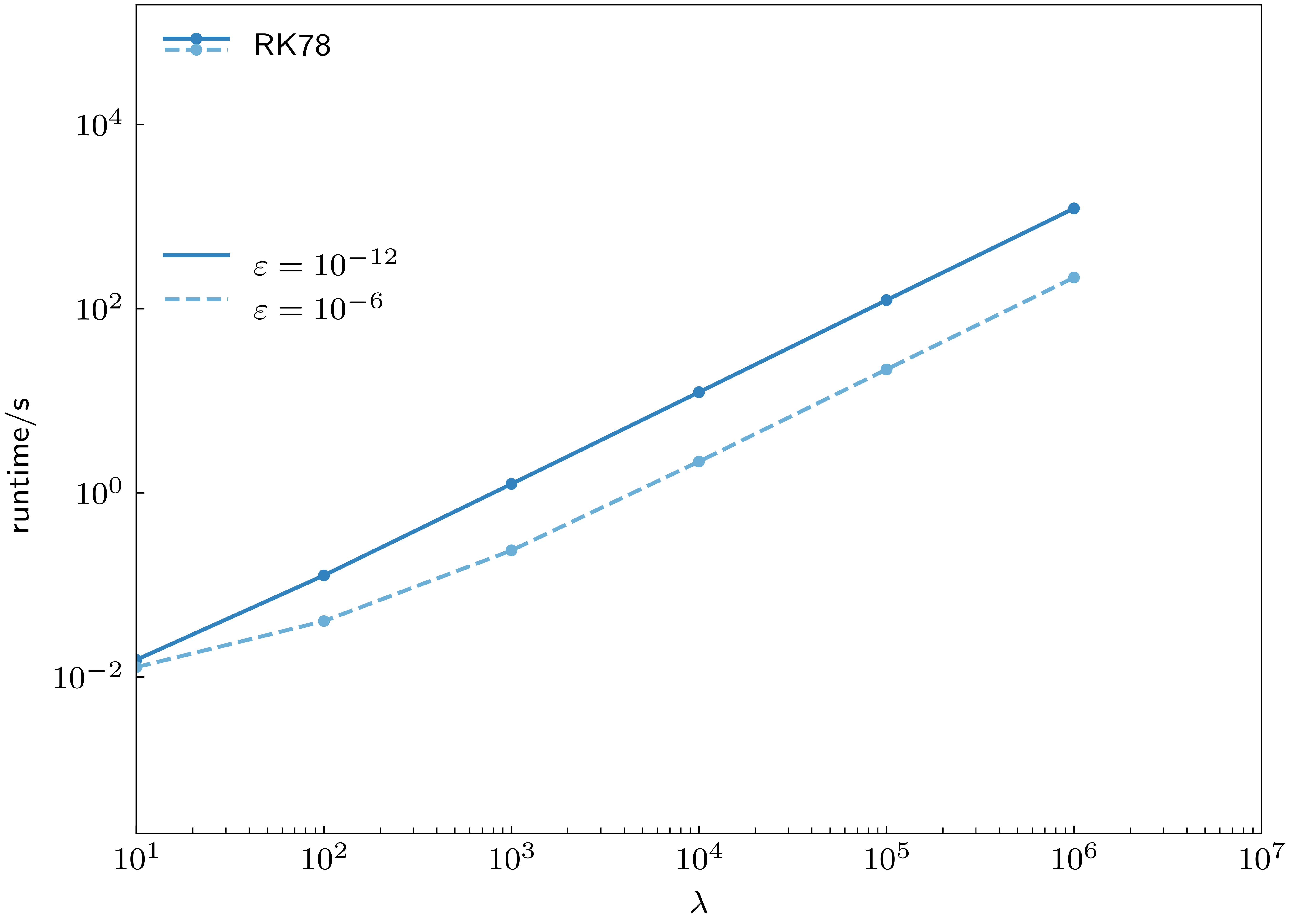
Example: $\lambda = 100$



Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

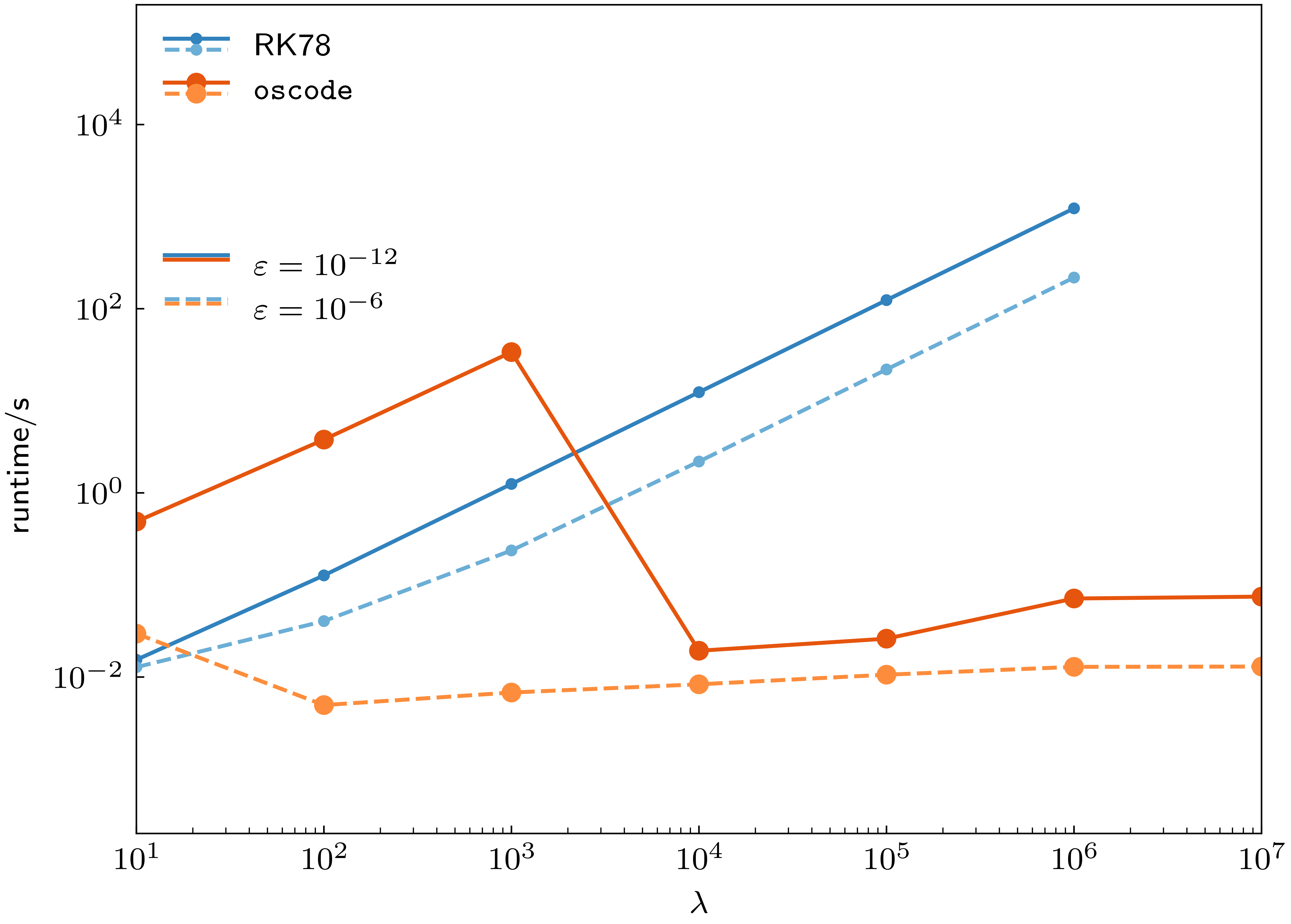
$$u''(t) + \lambda^2 q(t) u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2 \cos(3t)$$



Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

$$u''(t) + \lambda^2 q(t) u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2 \cos(3t)$$

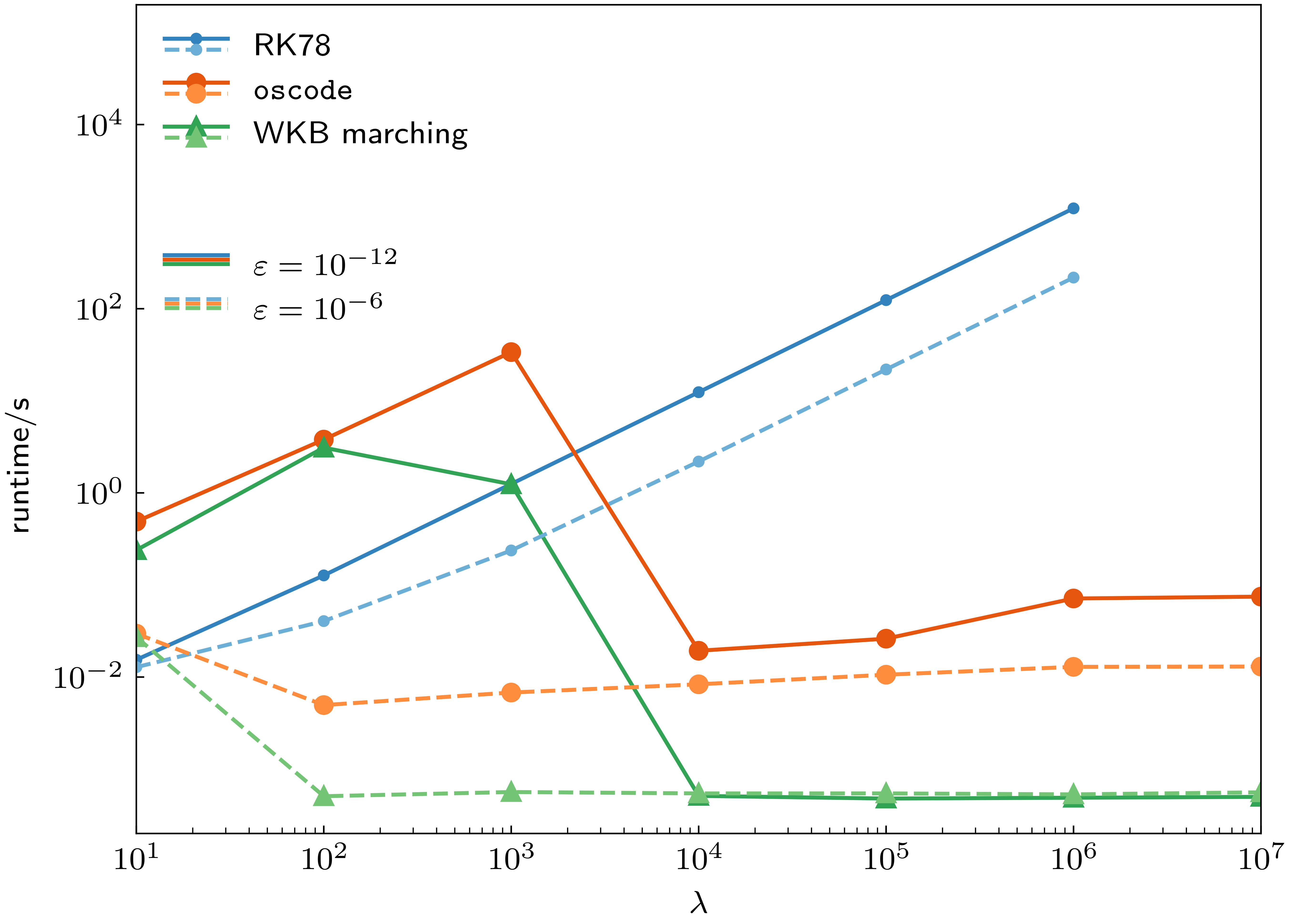


Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

$$u''(t) + \lambda^2 q(t) u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2 \cos(3t)$$

- WKB marching: *Körner et al, JCAM, 2022.*
  - Uses `oscode`'s switching algorithm
  - User needs to supply derivatives

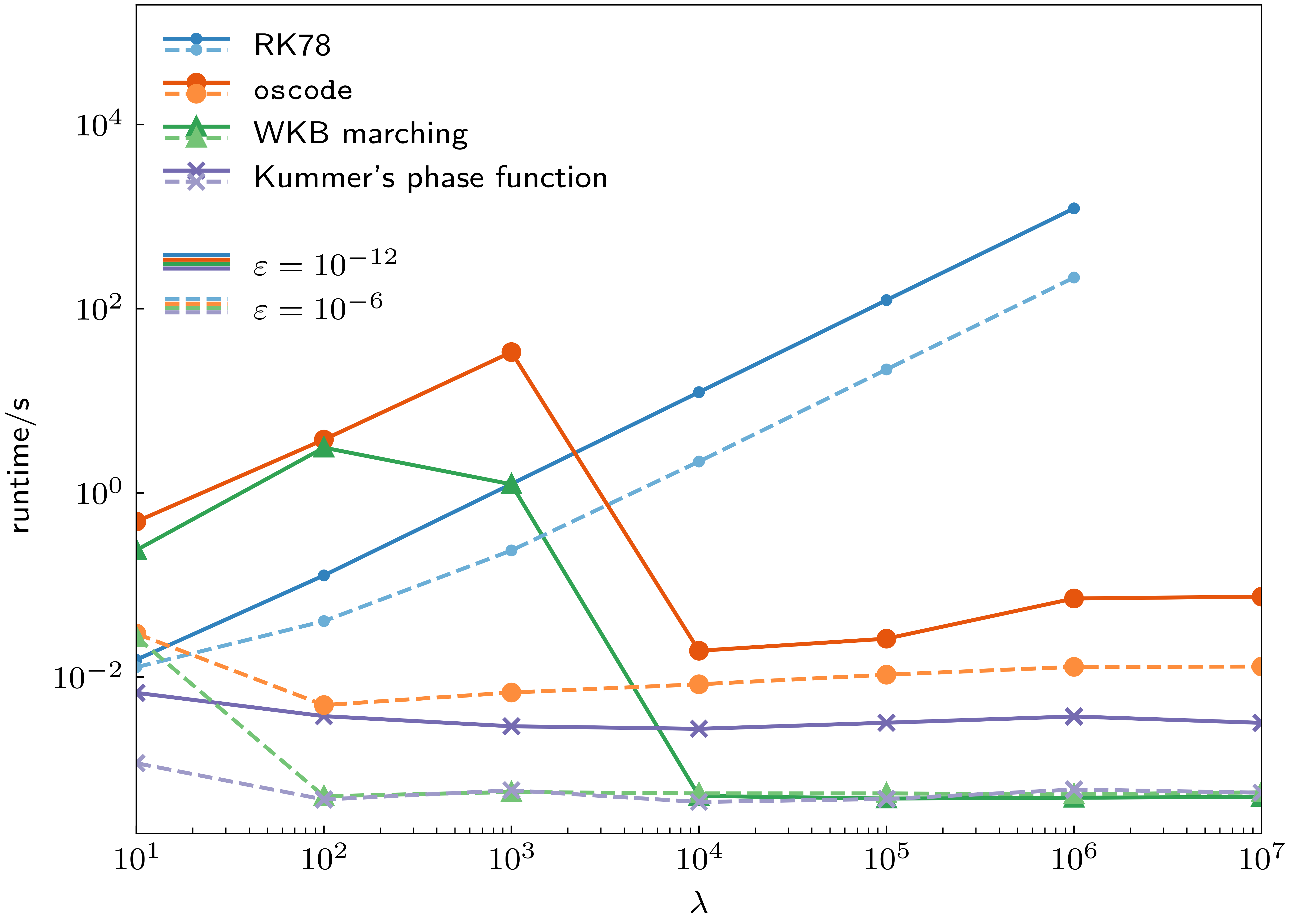


Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

$$u''(t) + \lambda^2 q(t) u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2 \cos(3t)$$

- WKB marching: *Körner et al, JCAM, 2022.*
  - Uses `oscode`'s switching algorithm
  - User needs to supply derivatives
- Kummer's phase function: *Bremer, ACHA, 2018.*
  - Only to be used in oscillatory regime

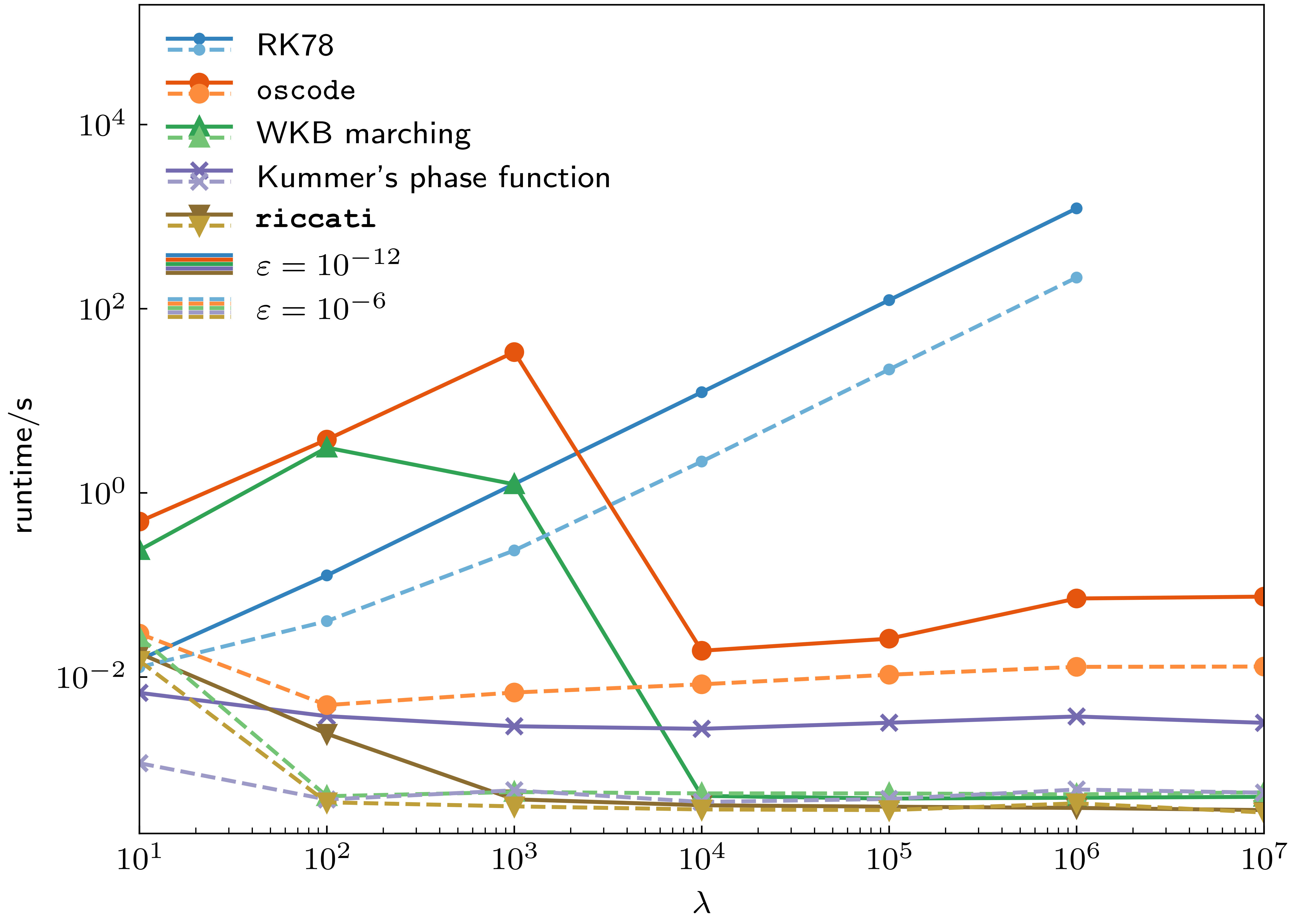


Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

$$u''(t) + \lambda^2 q(t)u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2\cos(3t)$$

- WKB marching: *Körner et al, JCAM, 2022.*
  - Uses `oscode`'s switching algorithm
  - User needs to supply derivatives
- Kummer's phase function: *Bremer, ACHA, 2018.*
  - Only to be used in oscillatory regime

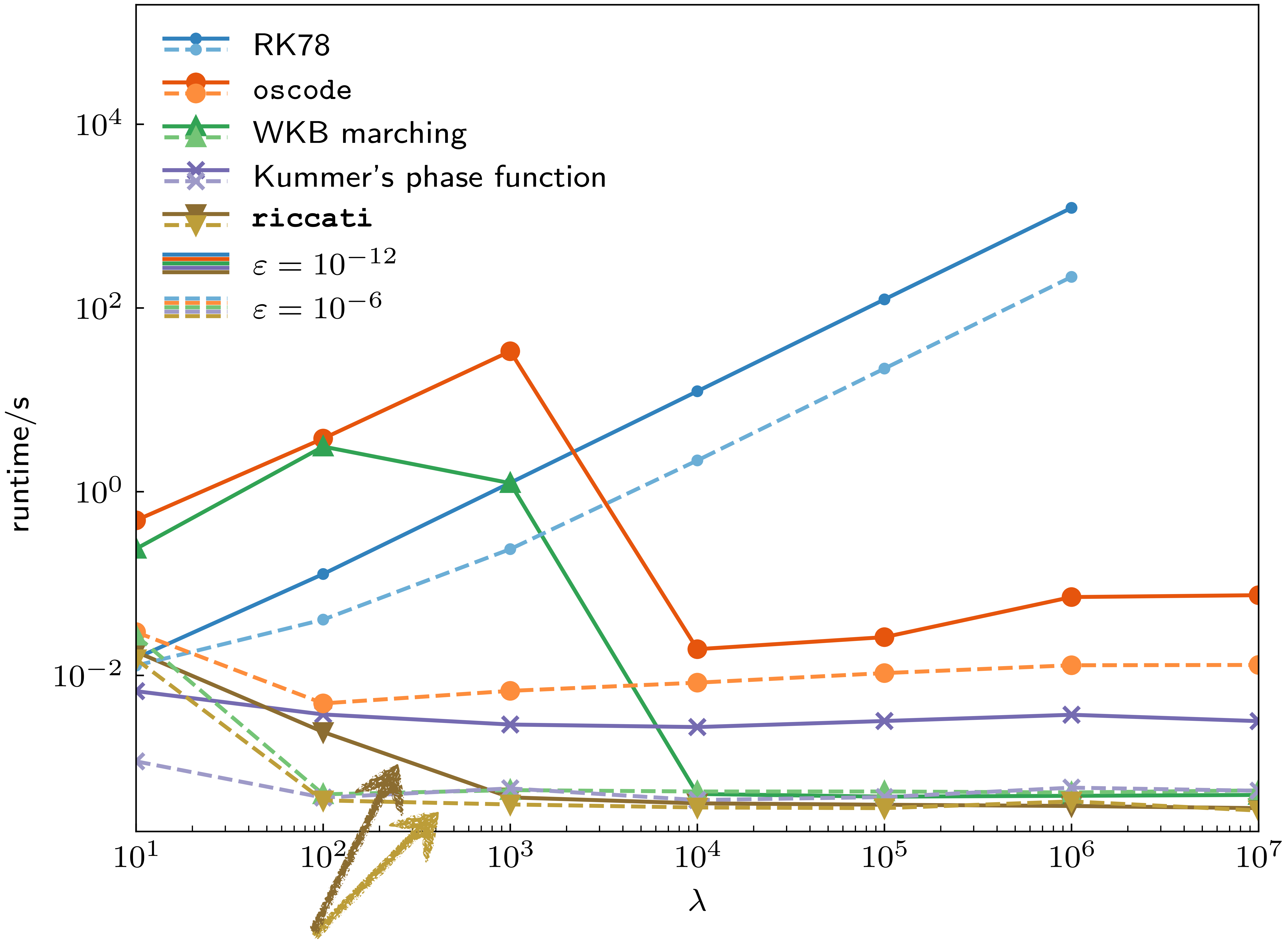


Example solution and solver comparison

- Compare state-of-the-art oscillatory solvers and a standard high-order Runge—Kutta method on:

$$u''(t) + \lambda^2 q(t)u(t) = 0,$$
$$t \in [-1,1],$$
$$q(t) = 1 - t^2\cos(3t)$$

- WKB marching: *Körner et al, JCAM, 2022.*
  - Uses `oscode`'s switching algorithm
  - User needs to supply derivatives
- Kummer's phase function: *Bremer, ACHA, 2018.*
  - Only to be used in oscillatory regime



Software

- Open-source, unit tested (with **continuous integration**), documented, with [executable tutorials]
- Easy install via `pip` or `conda(-forge)` (or from source)
- C++ header-only library + Python wrapper (`oscode`) or + pure Python (`riccati`)
- Both codes are **peer-reviewed** and published in JOSS (Journal of Open-Source Software)
- **`SUNDIALS`** integration in progress

riccati

riccati is a Python package implementing the adaptive Riccati defect correction (ARDC) method by Agocs & Barnett.

ARDC is a numerical method for solving ordinary diffential equations (ODEs) of the form

$$u''(t) + 2\gamma(t)u'(t) + \omega^2(t)u(t) = 0,$$

on some solution interval $t \in [t_0, t_1]$ and subject to the initial conditions $u(t_0) = u_0$, $u'(t_0) = u'_0$.

This documentation will show you how to use the package which is under active development on GitHub.

Shields

How to use the docs

Start by following the (brief) Installation guide. After that you may get started straight away with the Quick Start, or check out some more Examples. Each function in the module is documented in the API.

User Guide

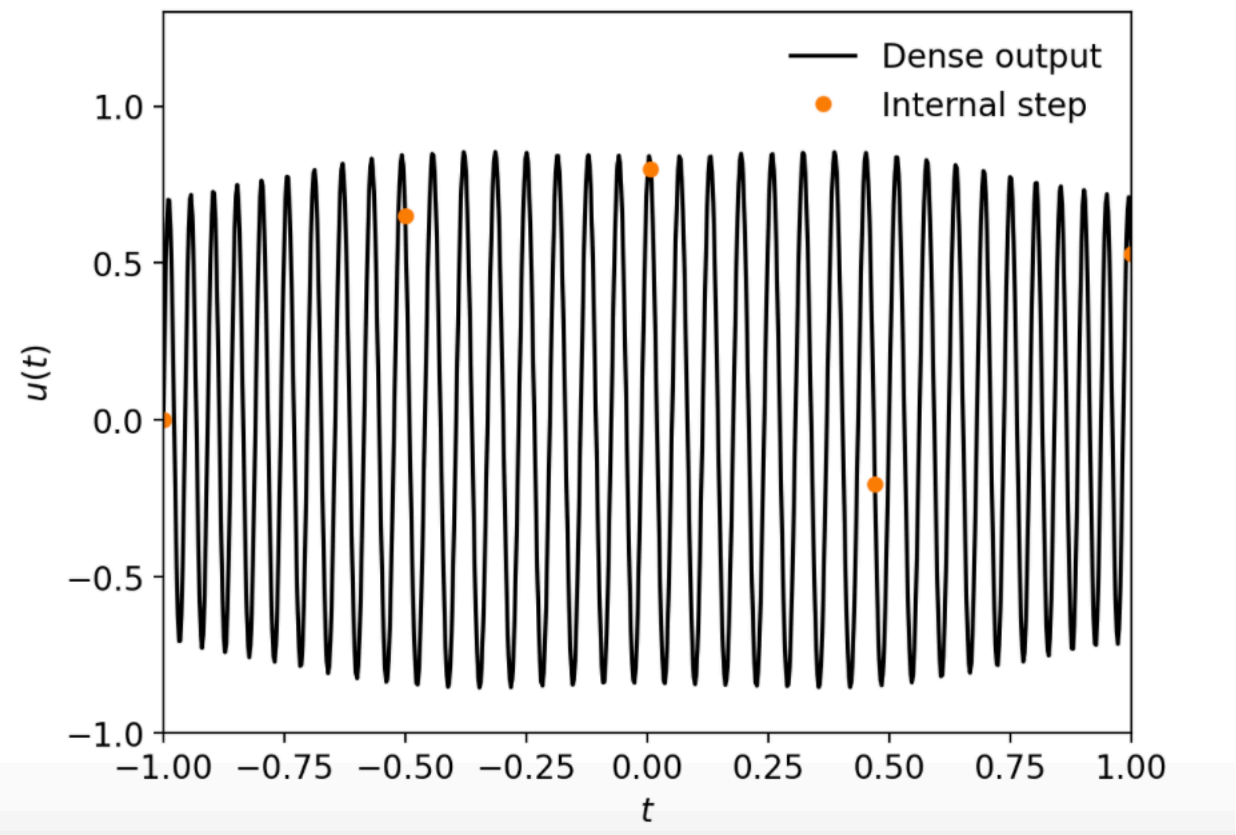
Basic example

We'll first demonstrate the basic functionality of the module with a minimal working example.

```
import numpy as np
import riccati

# Set up the ODE
w = lambda t: 100*np.sqrt(1 - t**2*np.cos(3*t))
g = lambda t: np.zeros_like(t) # Make sure the result is vectorised!

# Set up the solver
info = riccati.solversetup(w, g)

# Integration range and initial conditions
ti = -1.0
tf = 1.0
ui = 0.0
dui = 100.0

# We ask for the solution to be interpolated and output at the following "dense
t_eval = np.linspace(ti, tf, 800)

# Solve!
ts, ys, *misc, y_eval = riccati.solve(info, ti, tf, ui, dui, xeval = t_eval, ha
```

And plot the output:

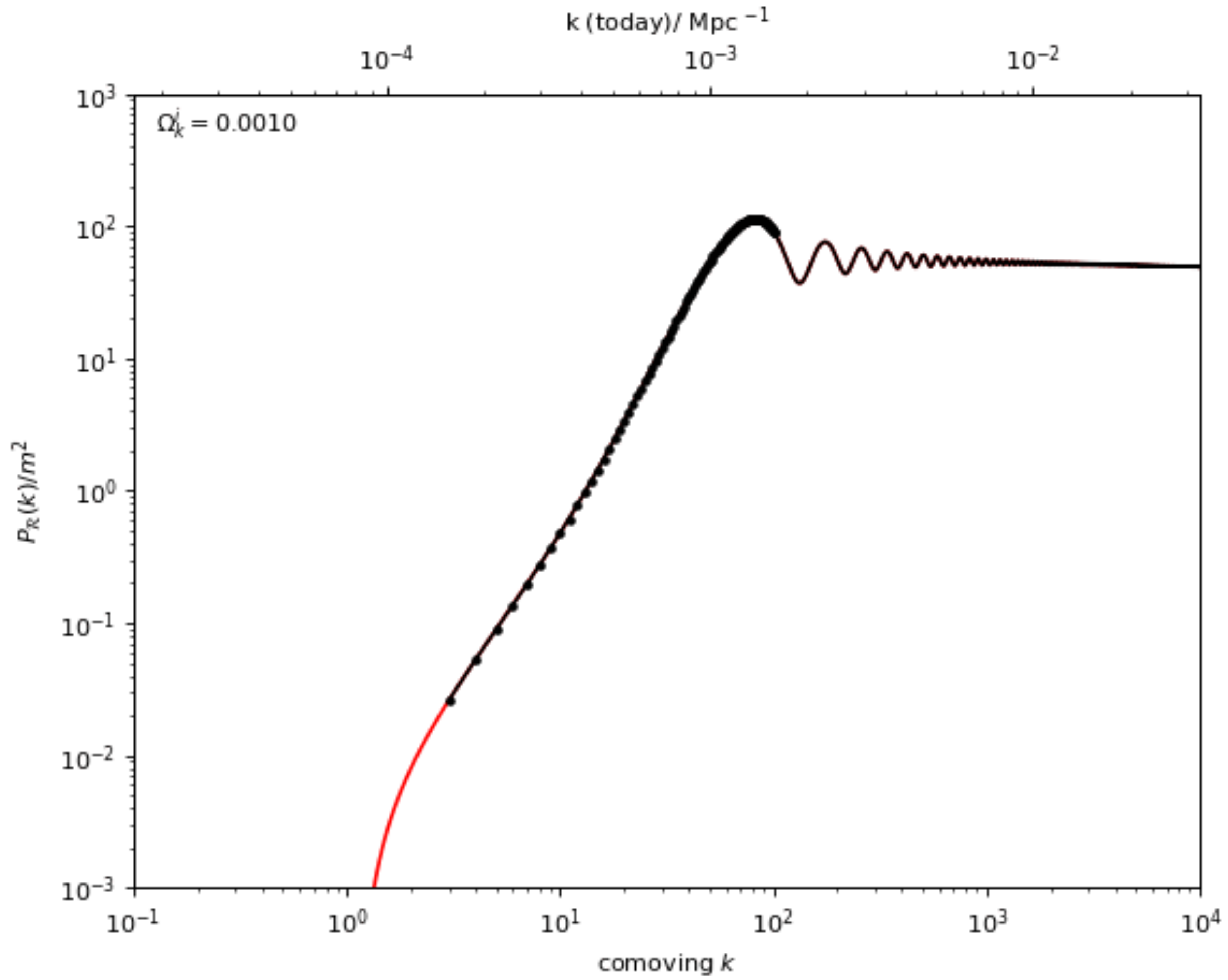
```
from matplotlib import pyplot as plt

plt.figure()
plt.plot(t_eval, y_eval, label = "Dense output", color = 'k')
plt.plot(ts, ys, '.', label = "Internal step", color = 'C1')
plt.xlim(ti, tf)
plt.ylim(-1, 1.3)
plt.xlabel('$t$')
plt.ylabel('$u(t)$')
plt.legend()
plt.show()
```

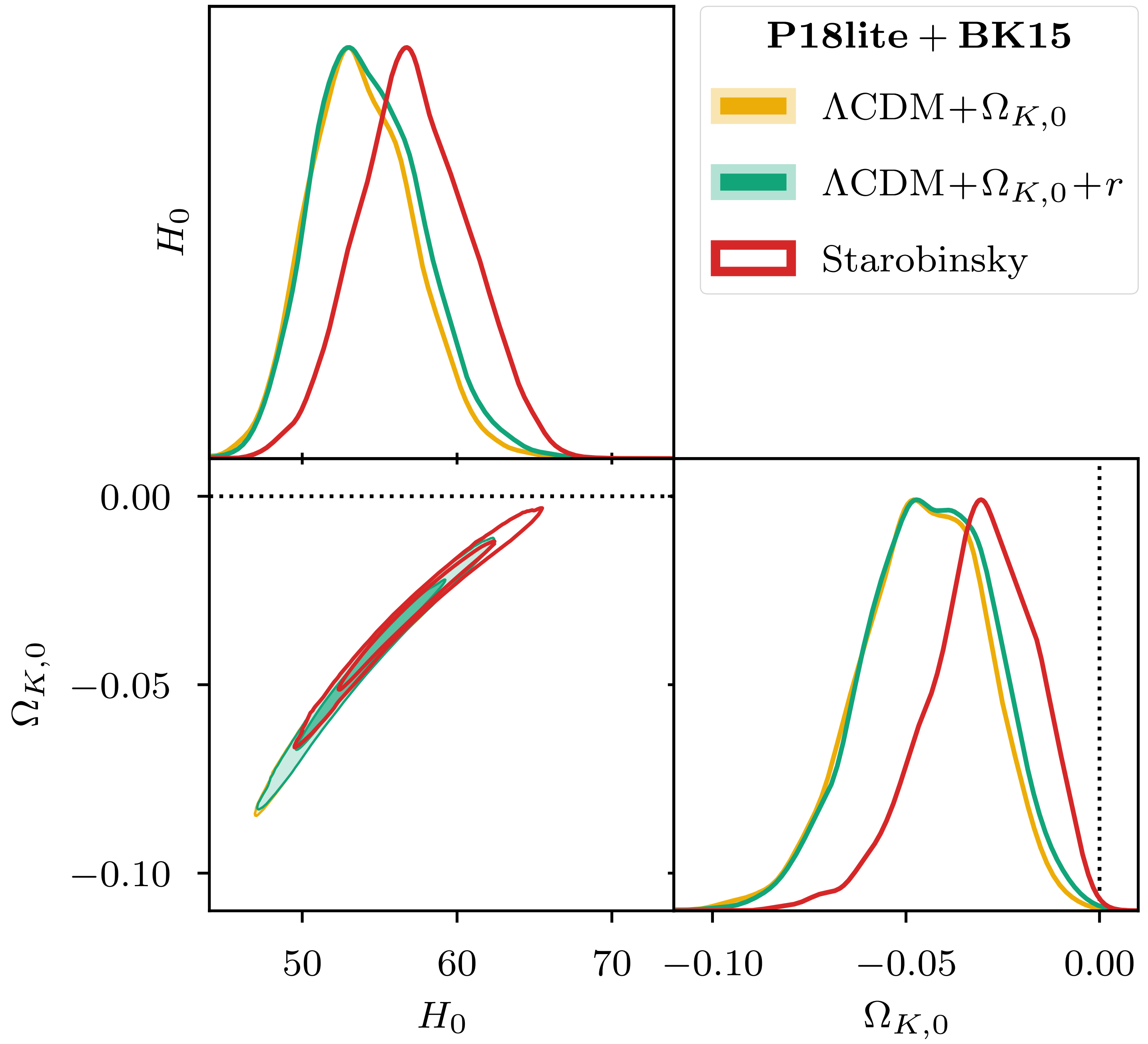


Applications & future work

Cosmology

- Forward-modeling step of CMB (Cosmic Microwave Background) involves $\approx 10^3$ osc. ODE solves, expensive → replaced with approximations
- Part of inference loop: 10^9 solves needed
- Computationally challenging models made possible to consider by `oscode`, `riccati`:
  - **curved (closed) universes** (currently favored by data): *Hergt, Agocs, et al. Phys Rev D, 2022.*
  - **Spectral distortion:** allows for particle physics scenarios during inflation, e.g. phase transitions, SUSY, strings





Applications & future work

Special function evaluation

- New fast solvers allow special functions to be evaluated by brute-force solving their ODE + interpolating (in terms of nonoscillatory phase function)
- Easier to parallelize for GPUs than currently preferred recursive formulae
- E.g. Associated Legendre functions, (spin) spherical harmonics, Wigner d functions

Timing results for evaluating Legendre polynomials via

$$(1-t^2)P_\nu'' - 2tP_\nu' + \nu(\nu+1)P_\nu = 0,$$

$$-1 < t < 1$$

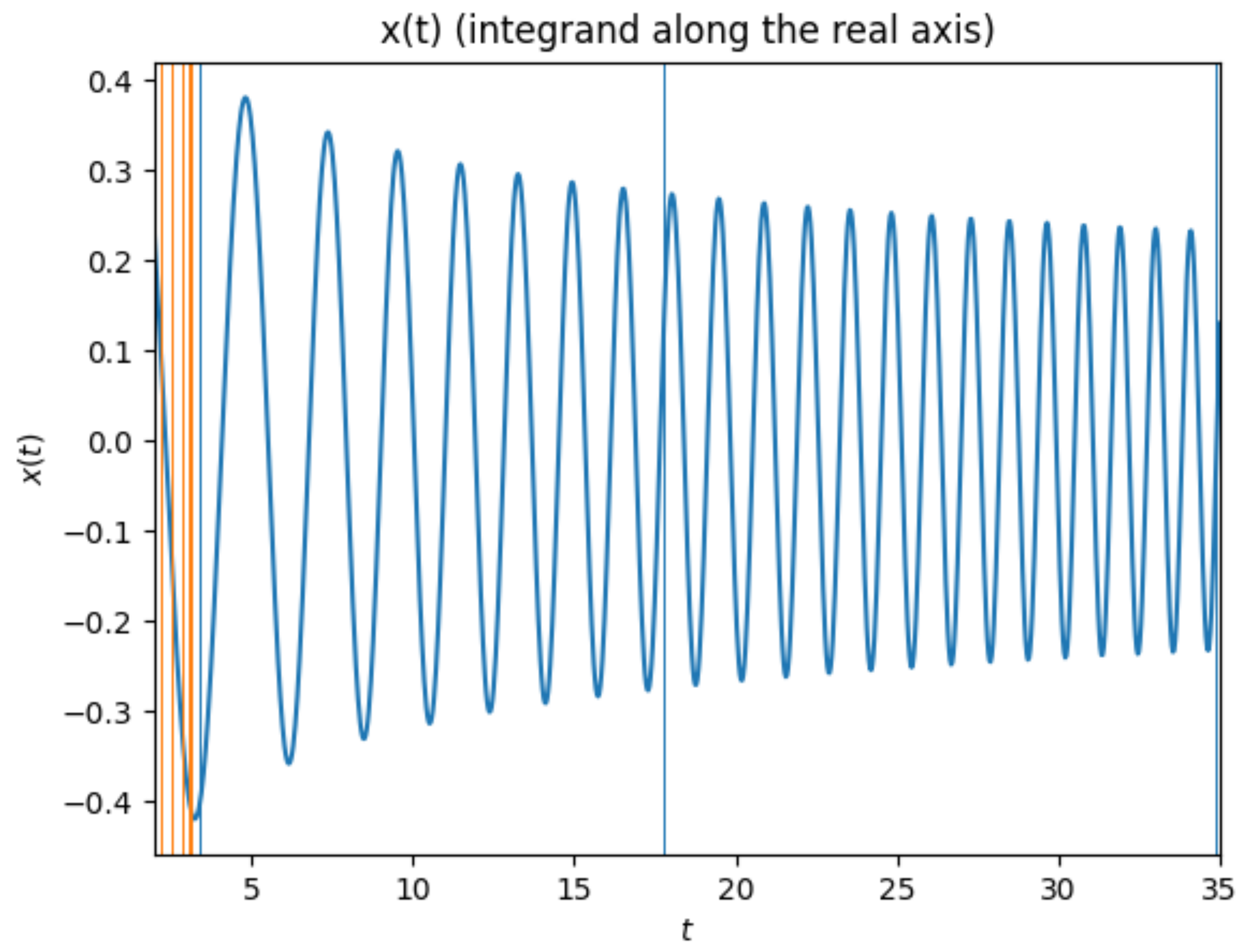
ν	Abs. error	ODE solve time/s	Func evals
10^1	8.76×10^{-12}	1.07×10^{-2}	5540
10^2	7.43×10^{-12}	3.88×10^{-2}	17840
10^3	3.51×10^{-13}	1.20×10^{-2}	5690
10^4	6.51×10^{-13}	5.54×10^{-3}	4108
10^5	2.06×10^{-12}	4.87×10^{-3}	4108
10^6	6.97×10^{-12}	4.55×10^{-3}	4108
10^7	1.76×10^{-11}	4.40×10^{-3}	4108
10^8	6.76×10^{-11}	4.25×10^{-3}	4108
10^9	2.06×10^{-10}	4.20×10^{-3}	4108

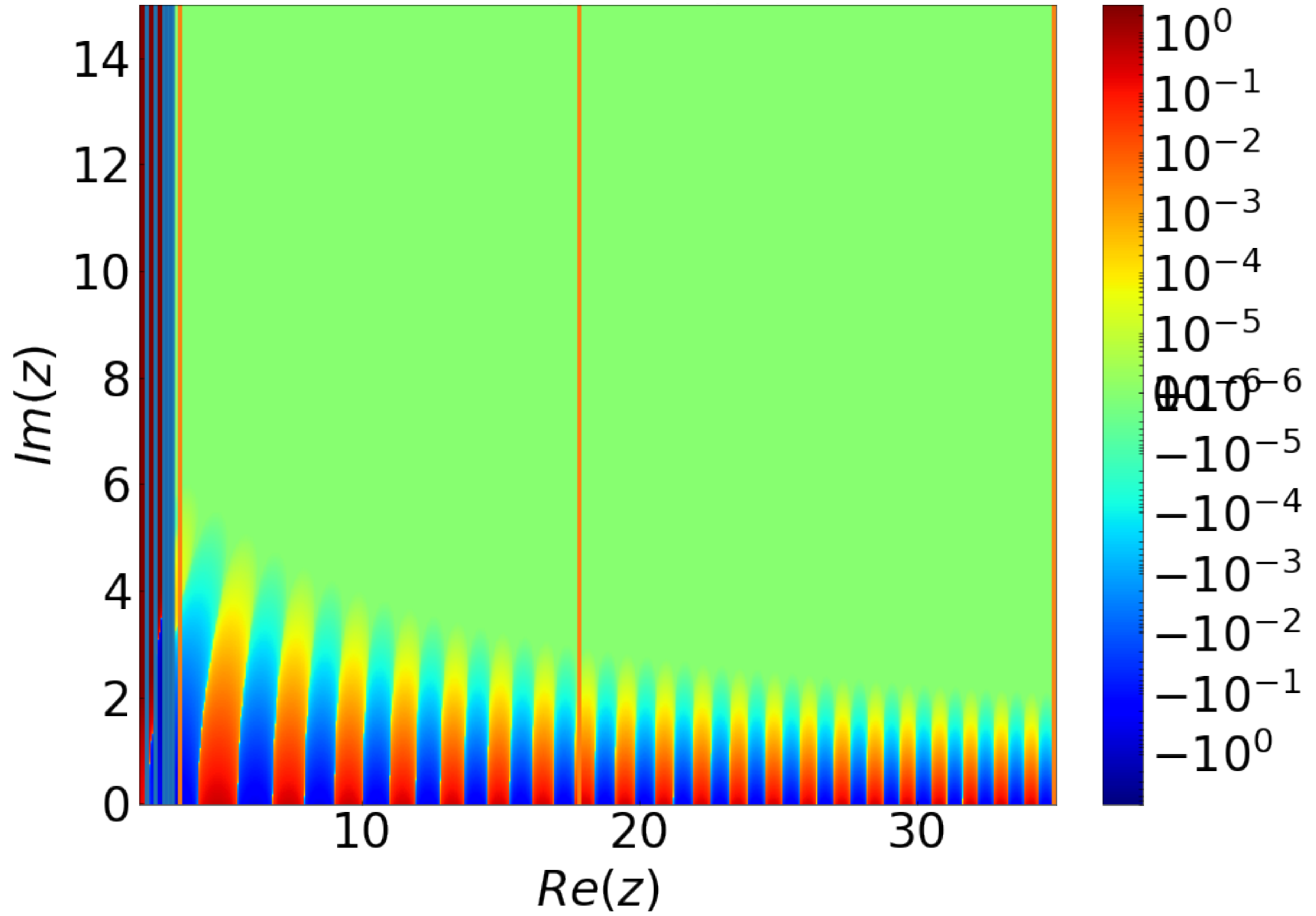
Each evaluation of $P_\nu(t)$ at a new t takes $\approx 10^{-6}$s

Applications & future work

Generalizations

- Quadrature of oscillatory functions
  - Represent an osc. function with its nonosc. phase function
  - Numerical steepest descent “goes around” the oscillations in the complex plane
  - **Gravitational wave template matching, CMB bispectrum calculation, wavefunction normalization**
- Nonlinear, oscillatory, second-order, homogeneous ODE
  - Inspired by Linda Petzold’s work: solve ODE system obeyed by Fourier coefficients
  - Numerical Poincaré—Lindstedt method
  - **Axion (dark matter candidate) realignment mechanism**
- Linear systems of coupled oscillatory ODEs





PDEs

Scattering of a nonperiodic source from a periodic, corrugated surface

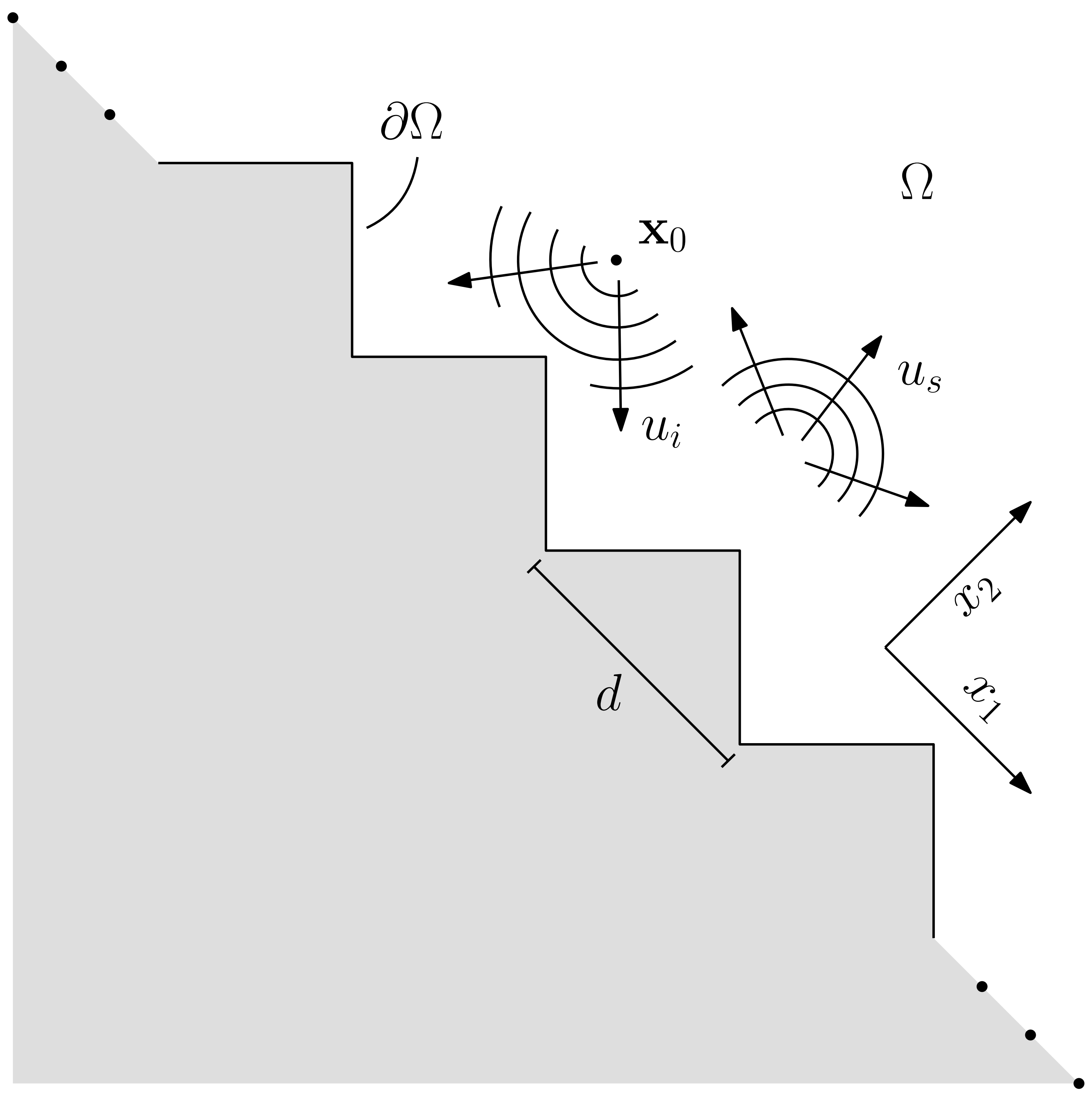
Why this problem?

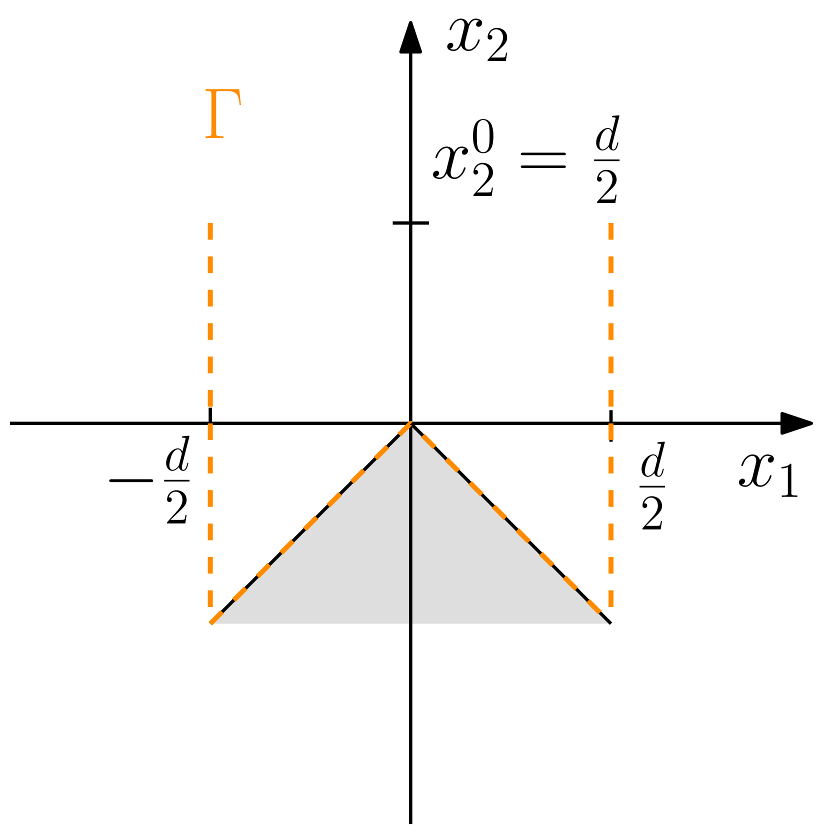
- Challenges:
  - Domain is infinite
  - Periodic boundary → cannot truncate due to artificial reflections
  - Nonperiodic source breaks periodicity → cannot reduce to single unit cell*
  - Corners introduce singularities
- Uses: waveguides, photonic crystals, acoustic metamaterials, diffraction gratings, antennae, anechoic chambers, amphitheaters, …
  - Fast, robust methods needed in **optimization** loops

What's novel?

- First **high-order accurate** scattering of a **nonperiodic source** from a **periodic surface with corners**: arXiv:2310.12486 (with Alex Barnett)
- Explains acoustic "raindrop" effect at pyramids via trapped acoustic modes
- Calculated power fraction transported away by trapped modes

Problem setup - quasiperiodic set of sources

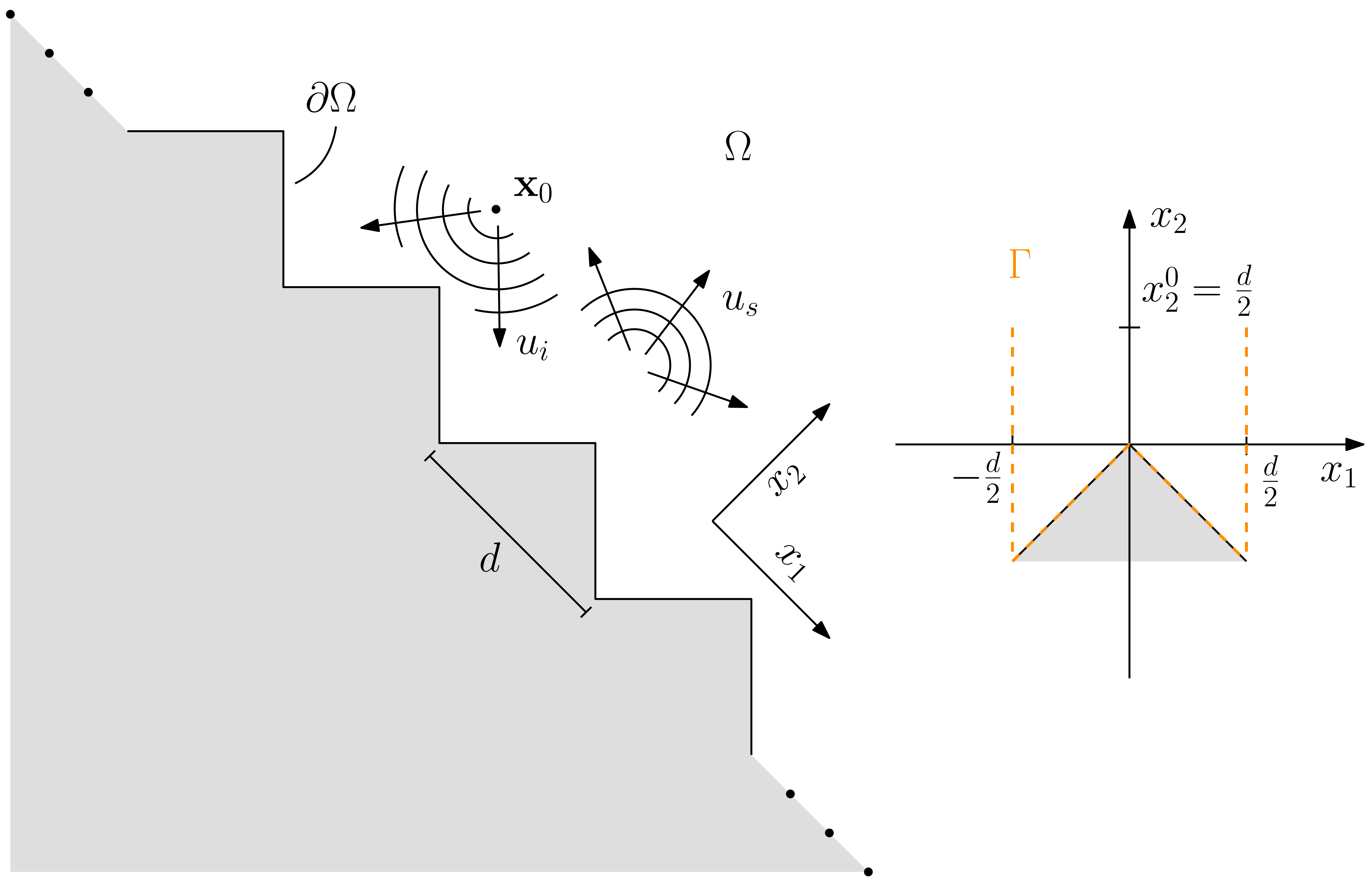




$-(\Delta + \omega^2)u = \sum_{n=-\infty}^{\infty} e^{in\kappa d}\delta(\mathbf{x} - \mathbf{x}_0 - n\mathbf{d})$	in Ω,	PDE (Helmholtz)
$u_n = 0$	on $\partial\Omega$,	boundary condition (Neu)
$u(x_1 + nd, x_2) = \alpha^n u(x_1, x_2)$	$(x_1, x_2) \in \Omega$,	quasiperiodicity
$u(x_1, x_2) = \sum_{n\in\mathbb{Z}} c_n e^{i(\kappa_n x_1 + k_n x_2)}$,	$x_2 > x_2^0$	radiation condition

- $\mathbf{x} = (x_1, x_2)$ position vector, $\mathbf{d} = (d, 0)$ lattice vector.
- $u = u_i + u_s$ is the total solution (incident + scattered)
- κ is the **horizontal (on-surface) wavenumber**
- $u_n := \mathbf{n} \cdot \nabla u$ normal derivative in the outward sense
- Solution accrues a **phase** $\alpha = e^{i\kappa d}$ over one **period** d. Quasiperiodicity condition
- Set of horizontal wavevectors $\kappa_n = \kappa + \frac{2\pi n}{d}$, $n \in \mathbb{Z}$, all equivalent
- If the total wavevector is $\mathbf{k} = (\kappa_n, k_n)$, then $k_n = \sqrt{\omega^2 - \kappa_n^2}$ is the vertical wavevector
  - Vertically propagating or evanescent
  - $k_n = 0$ are **Wood anomalies** (change in behavior)

Problem setup - quasiperiodic set of sources



- $\mathbf{x} = (x_1, x_2)$ position vector, $\mathbf{d} = (d, 0)$ lattice vector.
- $u = u_i + u_s$ is the total solution (incident + scattered)
- κ is the **horizontal (on-surface) wavenumber**
- $u_n := \mathbf{n} \cdot \nabla u$ normal derivative in the outward sense
- Solution accrues a **phase** $\alpha = e^{i\kappa d}$ over one **period** d. Quasiperiodicity condition
- Set of horizontal wavevectors $\kappa_n = \kappa + \dfrac{2\pi n}{d}$, $n \in \mathbb{Z}$, all equivalent
- If the total wavevector is $\mathbf{k} = (\kappa_n, k_n)$, then $k_n = \sqrt{\omega^2 - \kappa_n^2}$ is the vertical wavevector
  - Vertically propagating or evanescent
  - $k_n = 0$ are **Wood anomalies** (change in behavior)

Summary:

Start from a set of quasiperiodic point sources (of sound)

We can move by a period d, solution only changes by a complex phase.

System to solve is PDE + boundary condition + symmetry + far-away behavior

Boundary integral formulation; theory

- Use a single-layer potential (SLP) representation for the scattered wave:

$$u_s(\mathbf{x}) = \mathcal{S}\sigma = \int_\Gamma \Phi_\mathrm{p}(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\mathrm{d}s_\mathbf{y}, \quad \mathbf{x} \in \mathbb{R}^2,$$

ensures u will satisfy the PDE.

- Using the appropriate **jump relations**, this gives the Fredholm integral equation

$$(I - 2D^\mathrm{T})\sigma = 2(u_i)_n \quad \text{on } \Gamma,$$

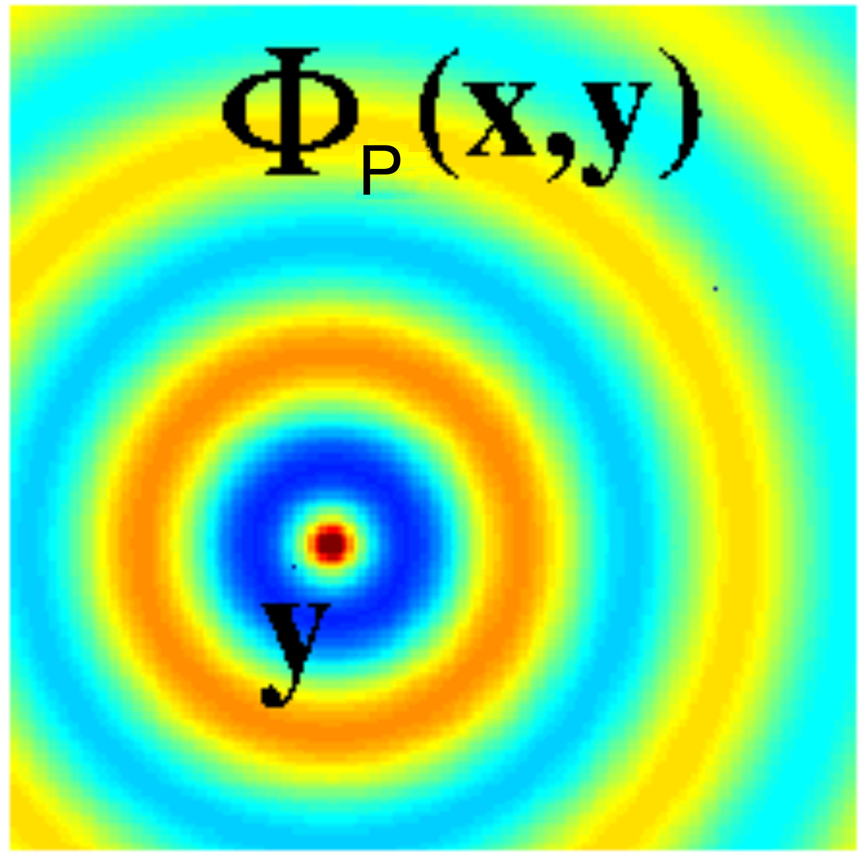
where σ is the unknown density, and

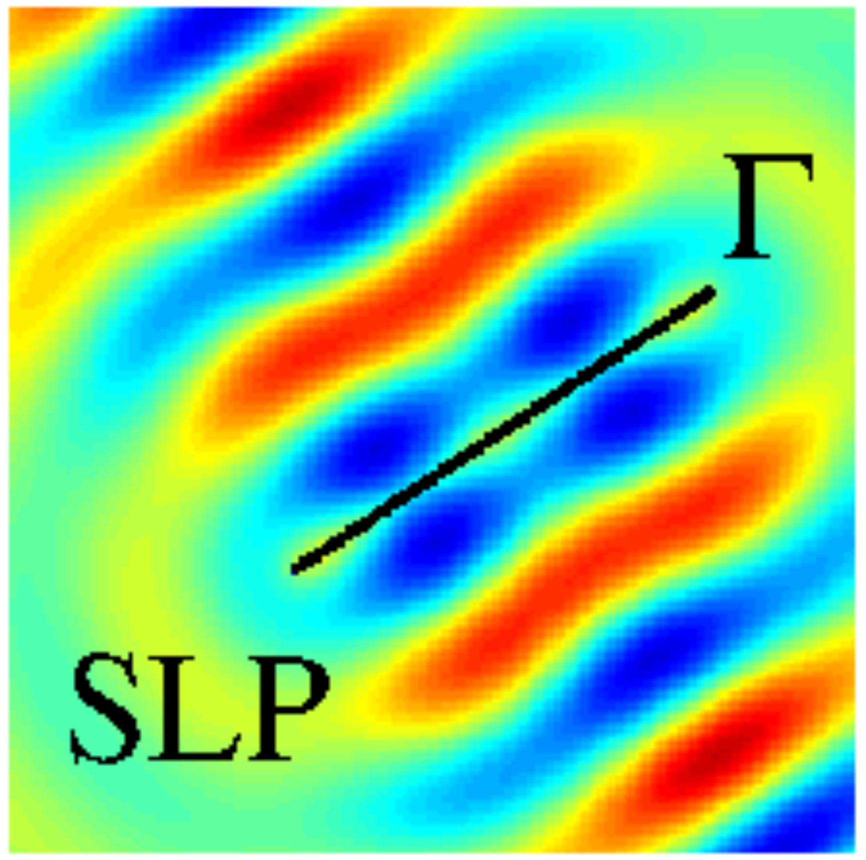
$$D^\mathrm{T}\sigma = \int_\Gamma \mathbf{n}_\mathbf{x} \cdot \nabla\Phi_\mathrm{p}(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\mathrm{d}s_\mathbf{y} \quad \text{on } \Gamma.$$

- Discretize via Nystrom's method, get dense linear system: $A\sigma = \mathbf{b}$
- Reconstruct solution u_s from σ via SLP everywhere
- $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$, can deal with singularities and be accurate via high-order quadrature

Quasiperiodic Green's function/fundamental solution:

- Solves PDE for quasiperiodic set of point sources
- Therefore depends on κ, ω
- Sum of Hankel functions





Boundary integral formulation; theory

- Use a single-layer potential (SLP) representation for the scattered wave:
$$u_s(\mathbf{x}) = \mathcal{S}\sigma = \int_\Gamma \Phi_\mathrm{p}(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\mathrm{d}s_\mathbf{y}, \quad \mathbf{x} \in \mathbb{R}^2,$$
ensures u will satisfy the PDE.
- Using the appropriate **jump relations**, this gives the Fredholm integral equation
$$(I - 2D^\mathrm{T})\sigma = 2(u_i)_n \quad \text{on } \Gamma,$$
where σ is the unknown density, and
$$D^\mathrm{T}\sigma = \int_\Gamma \mathbf{n}_\mathbf{x} \cdot \nabla\Phi_\mathrm{p}(\mathbf{x},\mathbf{y})\sigma(\mathbf{y})\mathrm{d}s_\mathbf{y} \quad \text{on } \Gamma.$$
- Discretize via Nystrom's method, get dense linear system: $A\sigma = \mathbf{b}$
- Reconstruct solution u_s from σ via SLP everywhere
- $\mathcal{O}(N)$ instead of $\mathcal{O}(N^2)$, can deal with singularities and be accurate via high-order quadrature

Quasiperiodic Green's function/fundamental solution:
- Solves PDE for quasiperiodic set of point sources
- Therefore depends on κ, ω
- Sum of Hankel functions

Summary:

Represent the solution with an ansatz.

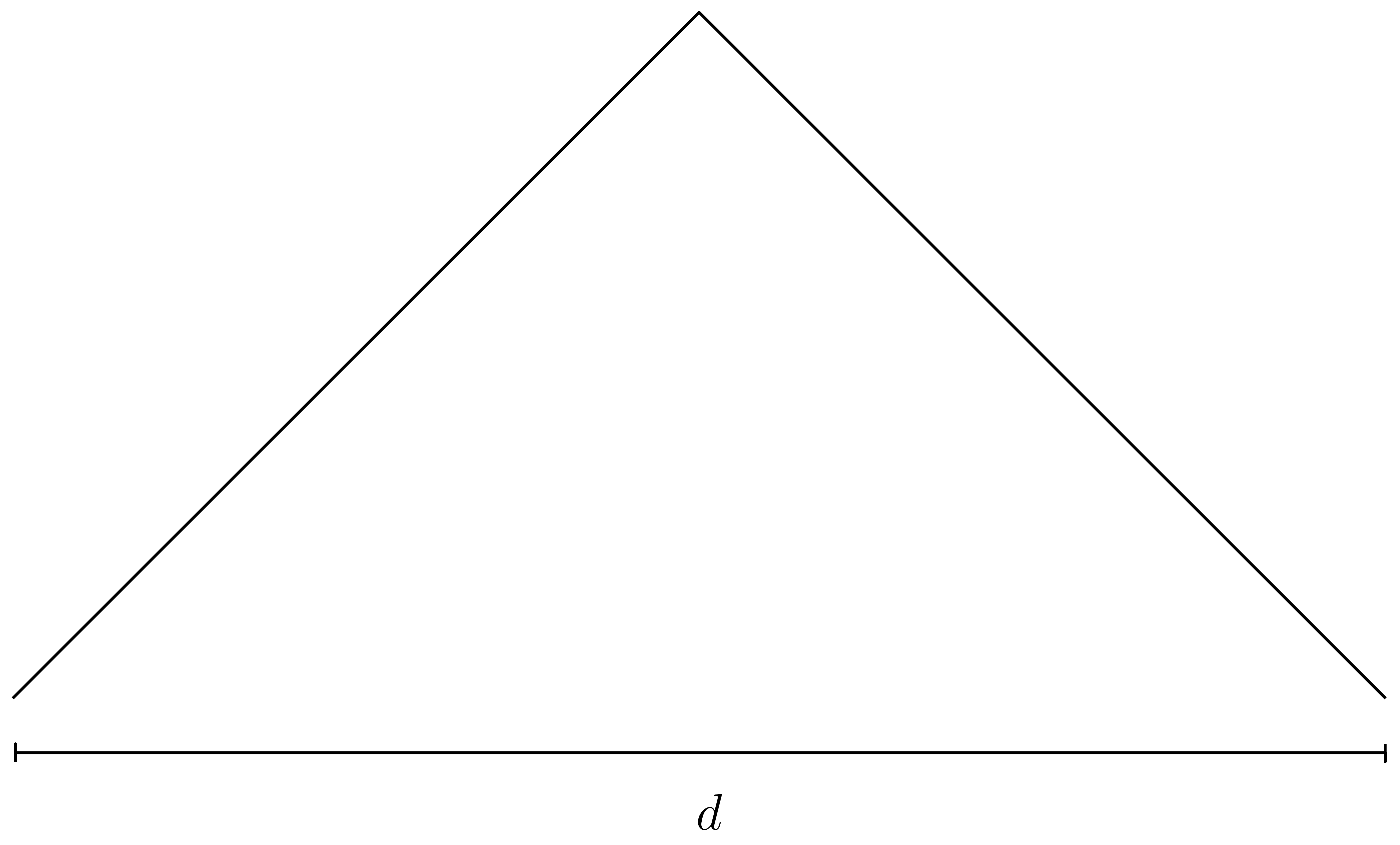
Ensuring the boundary conditions gives an integral equation that lives on the boundary

Discretize integral equation → dense linear system

We reduced the number of dimensions by 1

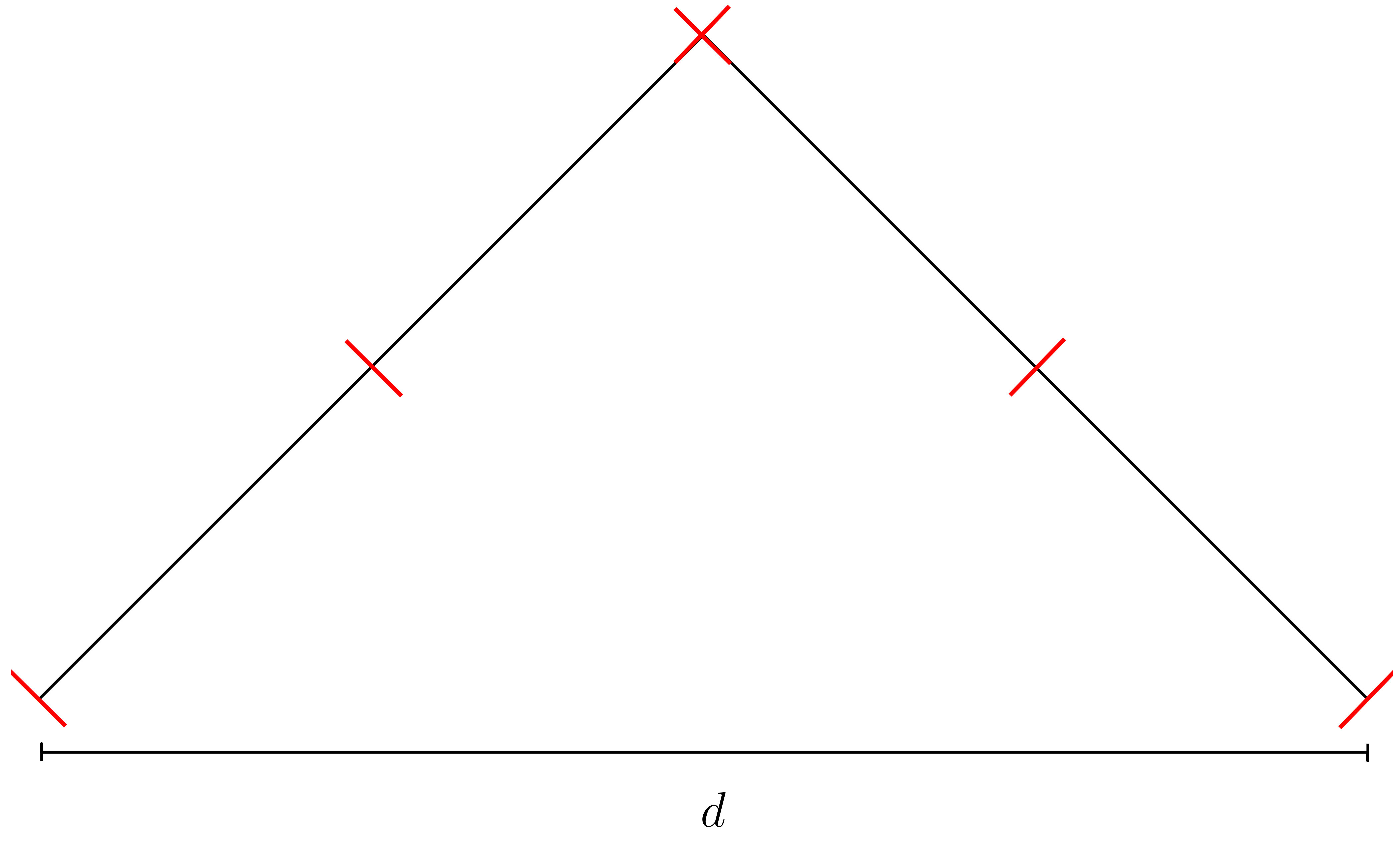
Boundary integral formulation; quadrature

- How to choose the quadrature nodes $\{s_i\}_{i=1}^{N}$?
- Integrand is singular at corners!
- $\rightarrow$ use panel quadrature with **adaptive corner refinement**:



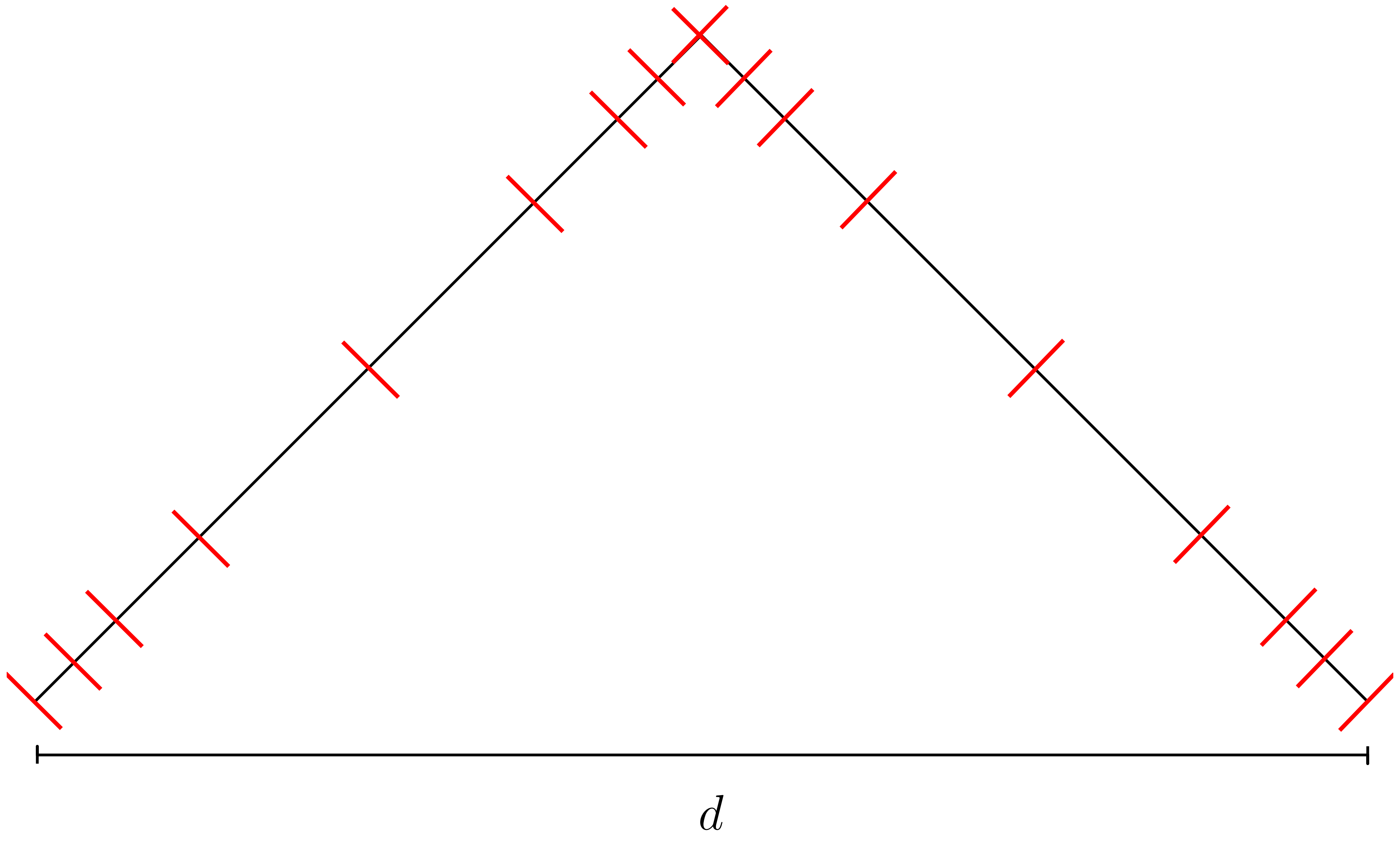
Boundary integral formulation; quadrature

- How to choose the quadrature nodes $\{s_i\}_{i=1}^{N}$?
- Integrand is singular at corners!
- → use panel quadrature with **adaptive corner refinement**:
    1. Lay down some equally sized initial panels



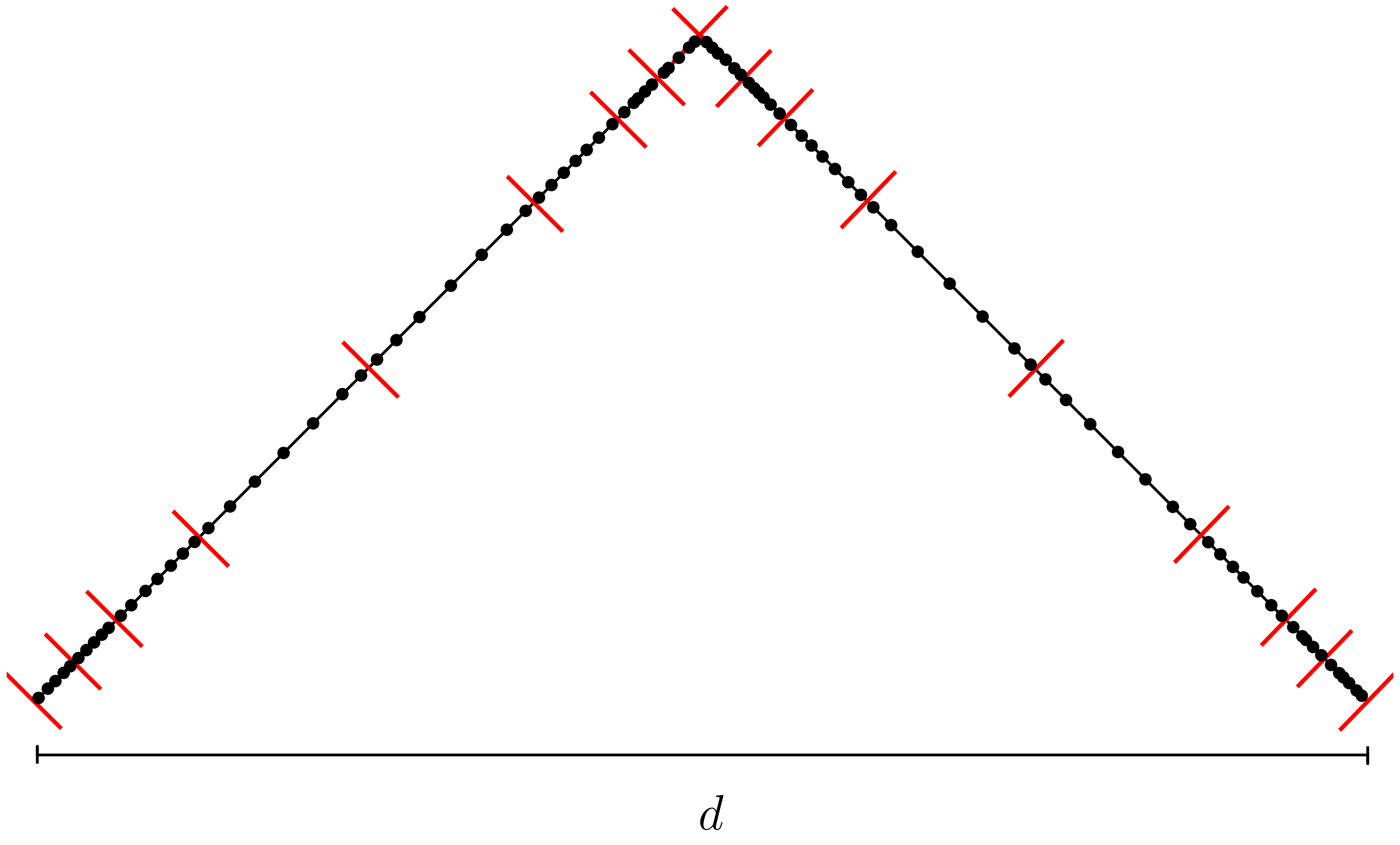
Boundary integral formulation; quadrature

- How to choose the quadrature nodes $\{s_i\}_{i=1}^{N}$?
- Integrand is singular at corners!
- $\rightarrow$ use panel quadrature with **adaptive corner refinement**:
    1. Lay down some equally sized initial panels
    2. Split corner-adjacent panels in a $1:(r-1)$ ratio ($r=2$, dyadic refinement shown)



Boundary integral formulation; quadrature

- How to choose the quadrature nodes $\{s_i\}_{i=1}^N$?
- Integrand is singular at corners!
- $\rightarrow$ use panel quadrature with **adaptive corner refinement**:
    1. Lay down some equally sized initial panels
    2. Split corner-adjacent panels in a $1:(r-1)$ ratio ($r=2$, dyadic refinement shown)
    3. Lay down **Gauss—Legendre** quadrature nodes on panels.
- Quadrature coordinates **relative** to the nearest corner to avoid catastrophic cancellation



Finding trapped modes, chirp reconstruction via ray model

- Trapped modes occur when the Fredholm determinant is singular, i.e.

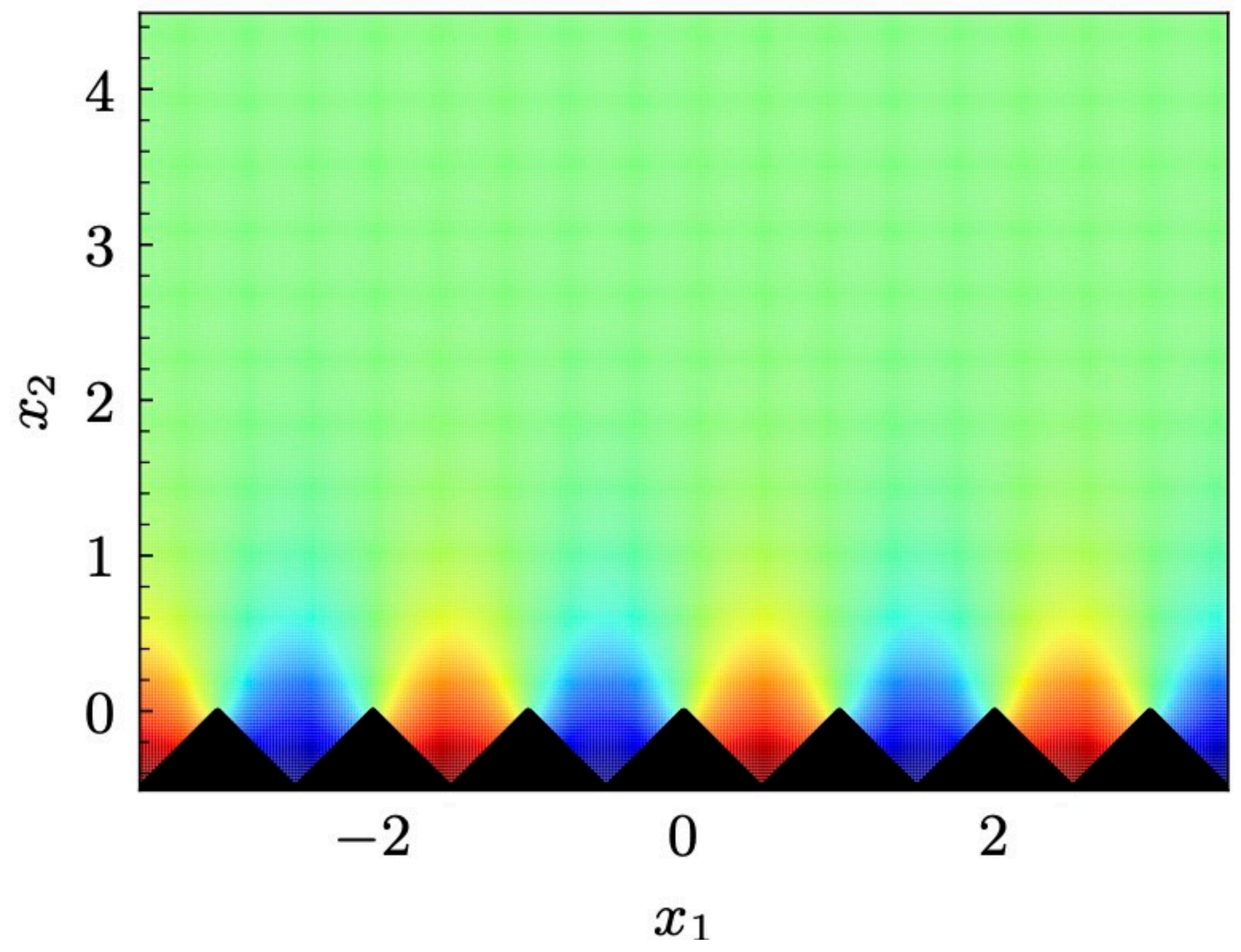
$$(I - 2D^{\mathrm{T}})\sigma = 0$$

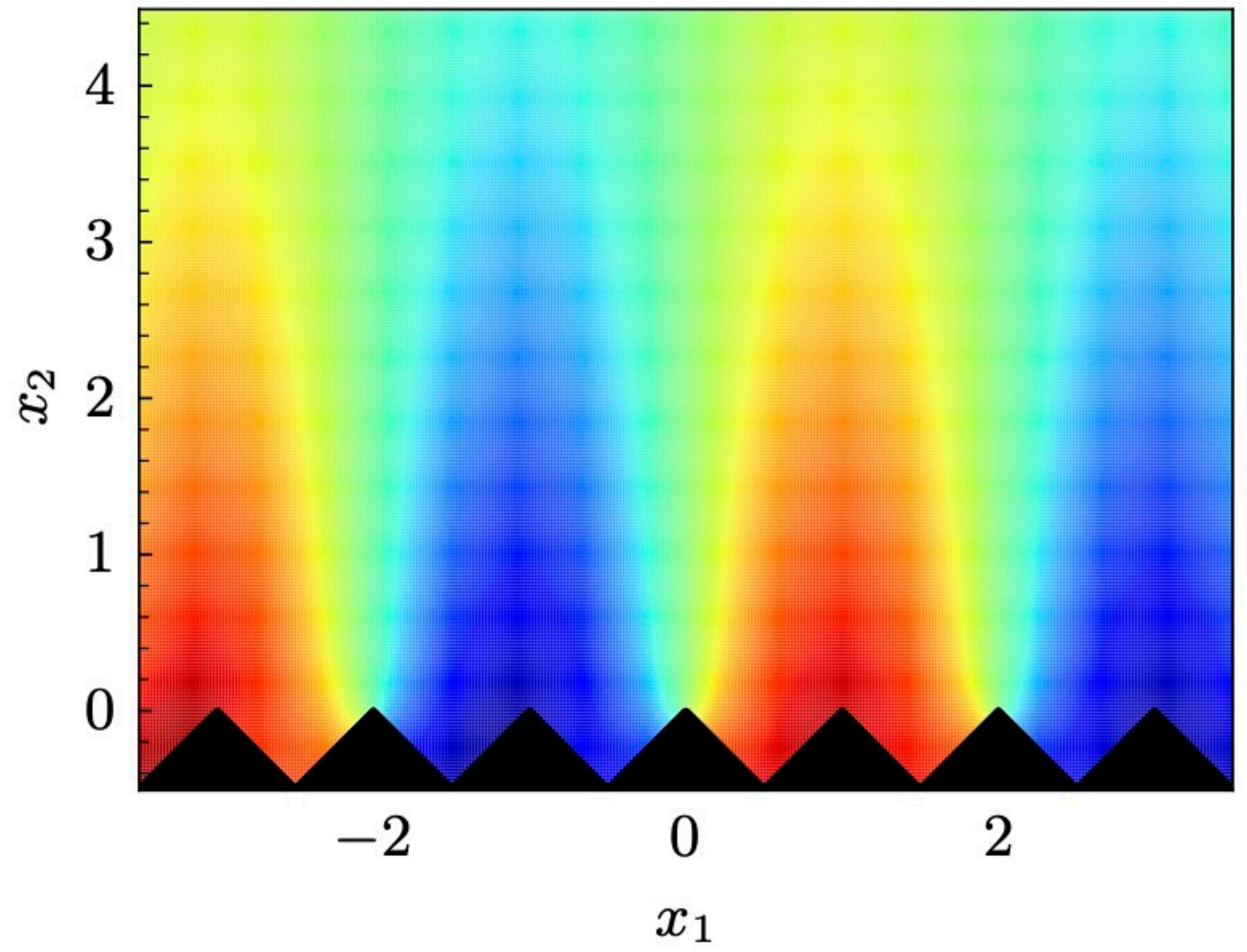
Recall that previously, we had $(I - 2D^{\mathrm{T}})\sigma = 2(u_i)_n$

has a nontrivial solution.

Finding trapped modes, chirp reconstruction via ray model

- Trapped modes occur when the Fredholm determinant is singular, i.e.
$$(I - 2D^{\mathrm{T}})\sigma = 0$$
has a nontrivial solution.
- Not a spurious resonance, this is a physical mode!
- D depends on κ, ω, so trapped modes only occur at some (κ, ω) combinations
- To find them: fix ω, sweep over all possible κ, $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$ and find roots of $\det(I - 2D^{\mathrm{T}})$ (with Newton's method)





Finding trapped modes, chirp reconstruction via ray model

- Trapped modes occur when the Fredholm determinant is singular, i.e.

$$(I - 2D^{\mathrm{T}})\sigma = 0$$

has a nontrivial solution.

- Not a spurious resonance, this is a physical mode!
- D depends on κ, ω, so trapped modes only occur at some (κ, ω) combinations
- To find them: fix ω, sweep over all possible κ, $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$ and find roots of $\det(I - 2D^{\mathrm{T}})$ (with Newton's method)

Recall that κ and $\kappa + \frac{2\pi n}{d}$ are equivalent

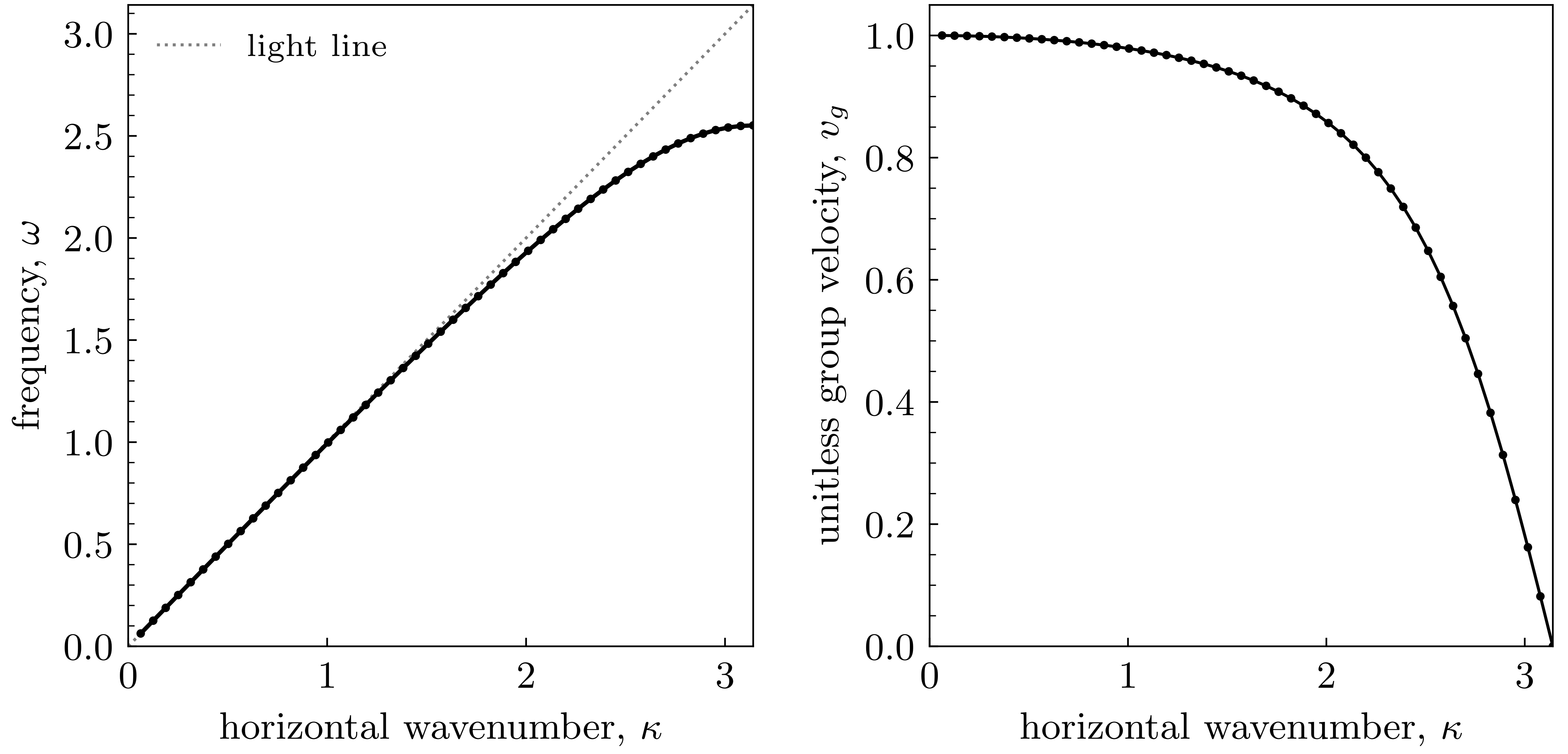
Finding trapped modes, chirp reconstruction via ray model

- Trapped modes occur when the Fredholm determinant is singular, i.e.

$$(I - 2D^{\mathrm{T}})\sigma = 0$$

  has a nontrivial solution.

- Not a spurious resonance, this is a physical mode!
- D depends on κ, ω, so trapped modes only occur at some (κ, ω) combinations
- To find them: fix ω, sweep over all possible κ, $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$ and find roots of $\det(I - 2D^{\mathrm{T}})$ (with Newton's method)
- Compute:
  - Dispersion relation, $\omega(\kappa)$, of trapped modes
  - The group velocity of a trapped mode, $\frac{\mathrm{d}\omega}{\mathrm{d}\kappa}$, velocity at which the envelope of a wavepacket travels



Finding trapped modes, chirp reconstruction via ray model

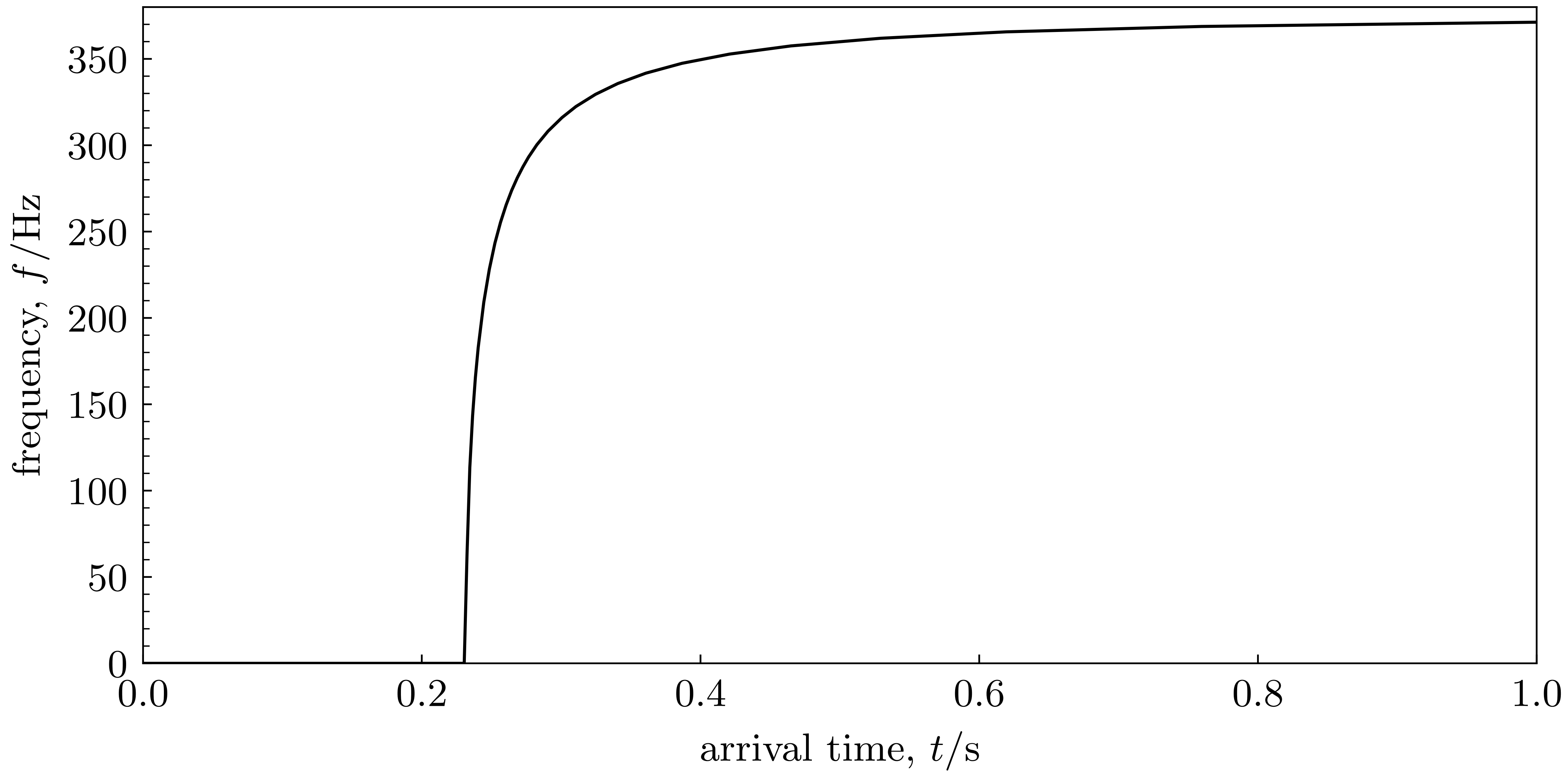
- Trapped modes occur when the Fredholm determinant is singular, i.e.

$$(I - 2D^{\mathrm{T}})\sigma = 0$$

has a nontrivial solution.

- Not a spurious resonance, this is a physical mode!
- D depends on κ, ω, so trapped modes only occur at some (κ, ω) combinations
- To find them: fix ω, sweep over all possible κ, $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$ and find roots of $\det(I - 2D^{\mathrm{T}})$ (with Newton's method)
- Compute:
  - Dispersion relation, $\omega(\kappa)$, of trapped modes
  - The group velocity of a trapped mode, $\frac{\mathrm{d}\omega}{\mathrm{d}\kappa}$, velocity at which the envelope of a wavepacket travels
  - **Ray model**: arrival time of different frequencies at El Castillo
    - Neglect: spreading along stairs in 3rd dimension; changes in amplitude; assume all trapped modes are excited

"Chirp" signal



Array scanning / Floquet—Bloch transform

- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ

Recall $\mathbf{d} = (d,0)$ is the lattice vector

$$\delta(\mathbf{x} - \mathbf{x}_0) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} \sum_{n=-\infty}^{\infty} e^{in\kappa d}\delta(\mathbf{x} - \mathbf{x}_0 - n\mathbf{d})\mathrm{d}\kappa,$$

$\rightarrow$ solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)

Array scanning / Floquet—Bloch transform

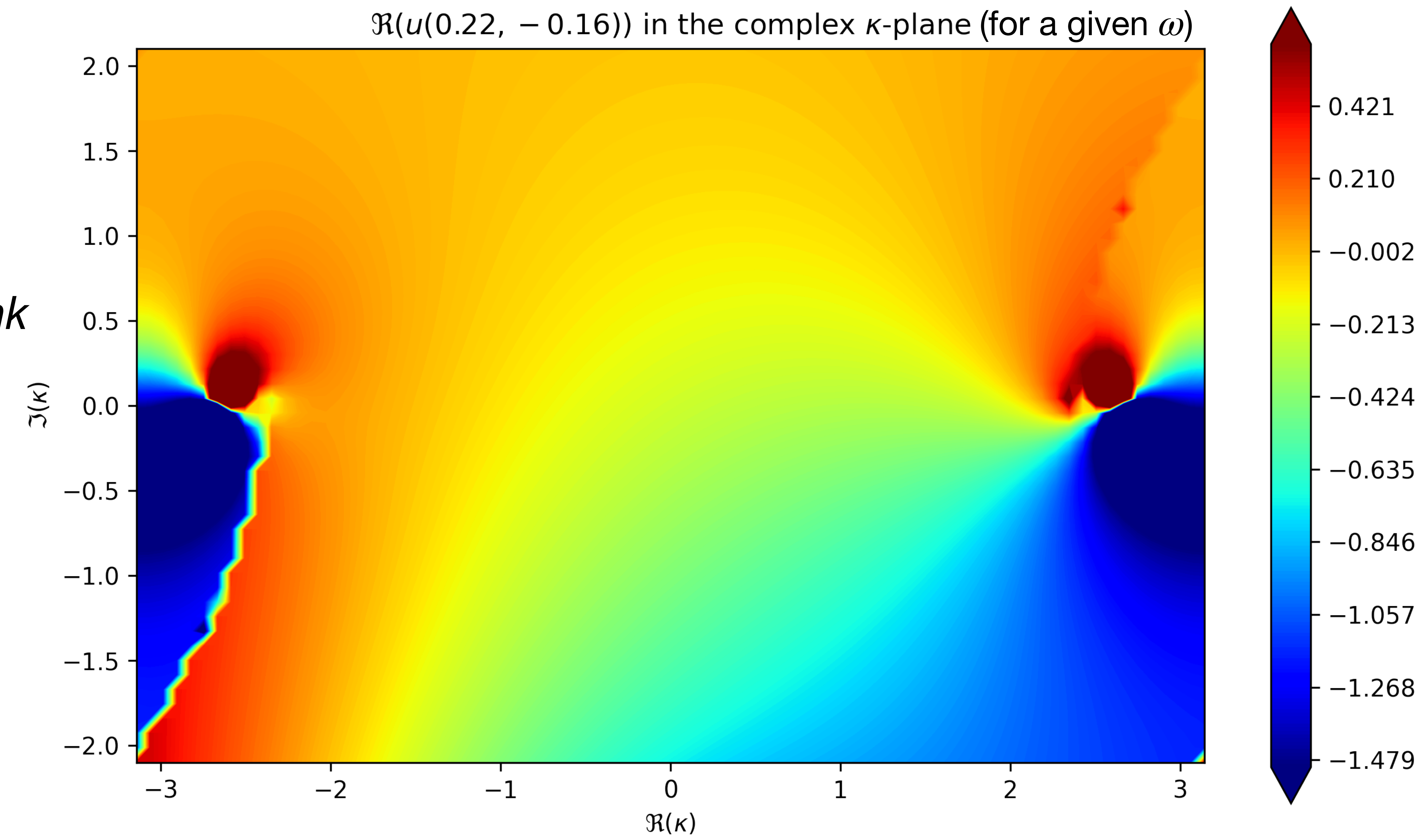
- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ
$$u(\mathbf{x}) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})\mathrm{d}\kappa,$$
$\rightarrow$ solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)

Array scanning / Floquet—Bloch transform

- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ

$$u(\mathbf{x}) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})\mathrm{d}\kappa,$$

$\rightarrow$ solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)



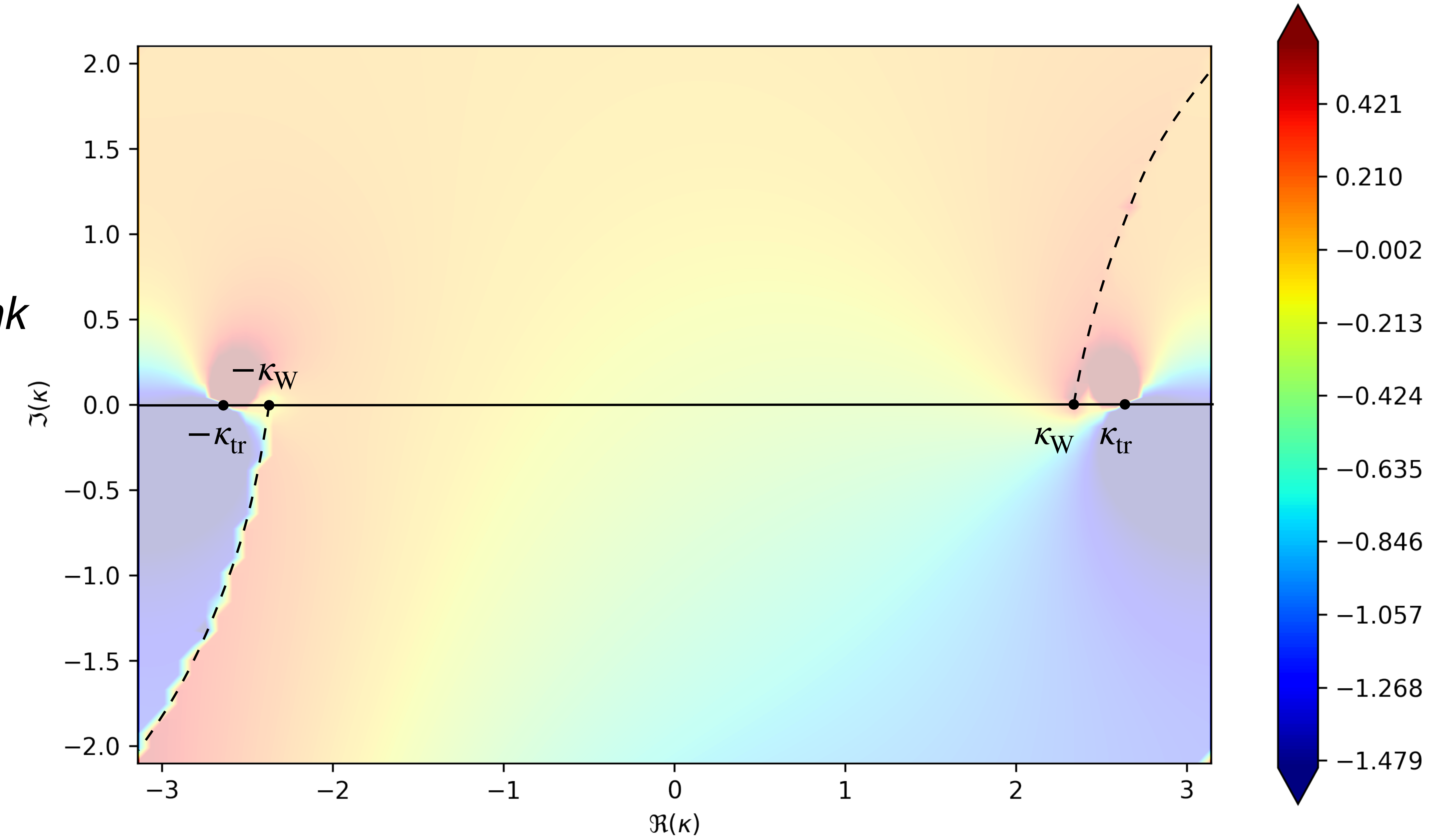
Array scanning / Floquet—Bloch transform

- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ

$$u(\mathbf{x}) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})\mathrm{d}\kappa,$$

$\rightarrow$ solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)

- But, on real axis:
  - Branch cuts at Wood anomalies κ_W, squareroot singularity
  - Poles at trapped modes κ_tr



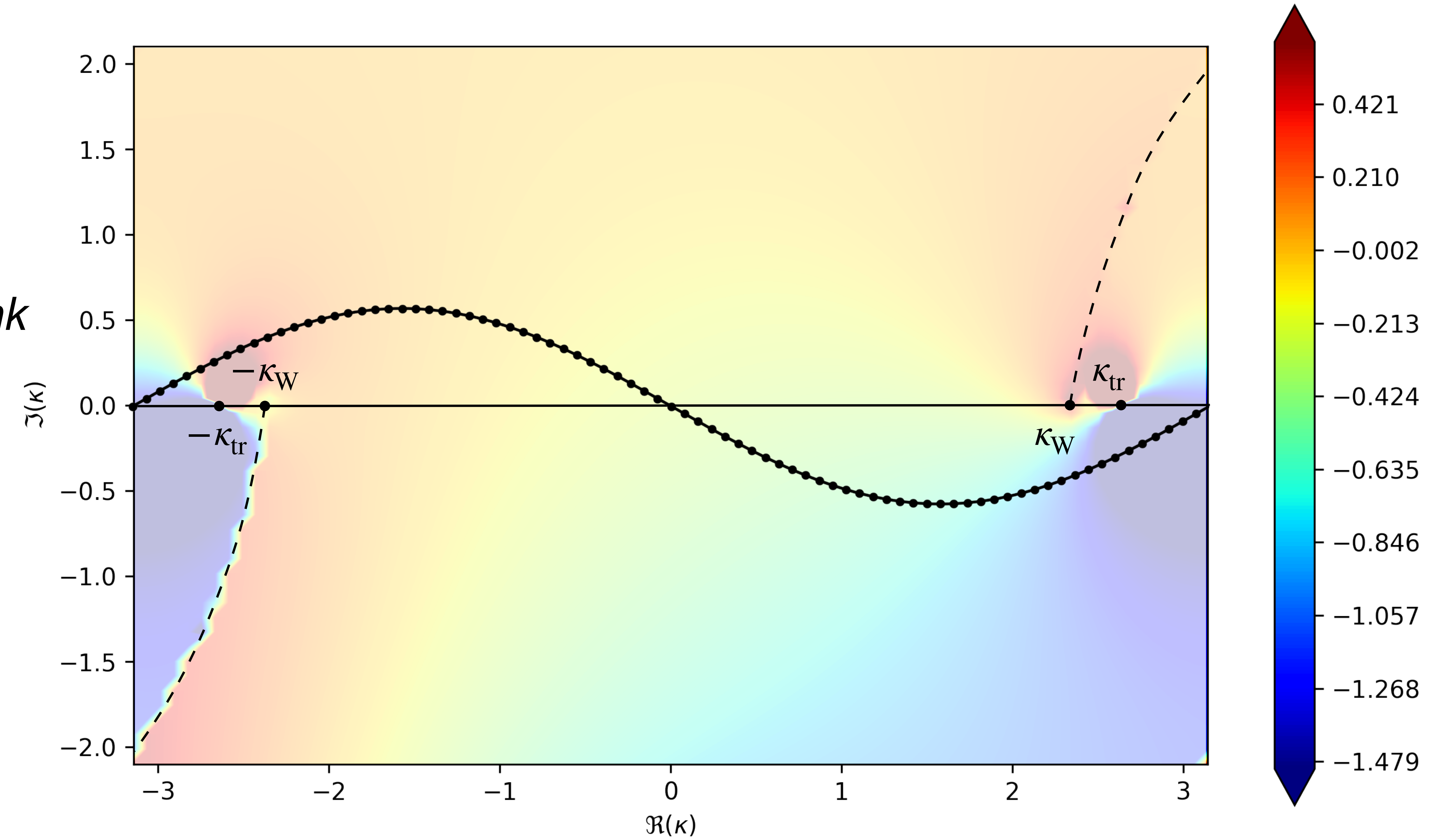
Array scanning / Floquet—Bloch transform

- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ

$$u(\mathbf{x}) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})\mathrm{d}\kappa,$$

  → solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)

- But, on real axis:
  - Branch cuts at Wood anomalies κ_W, squareroot singularity
  - Poles at trapped modes κ_tr
- Contour deformation (example path shown), sinusoidal with amplitude A, trapezoidal rule with P_asm nodes*



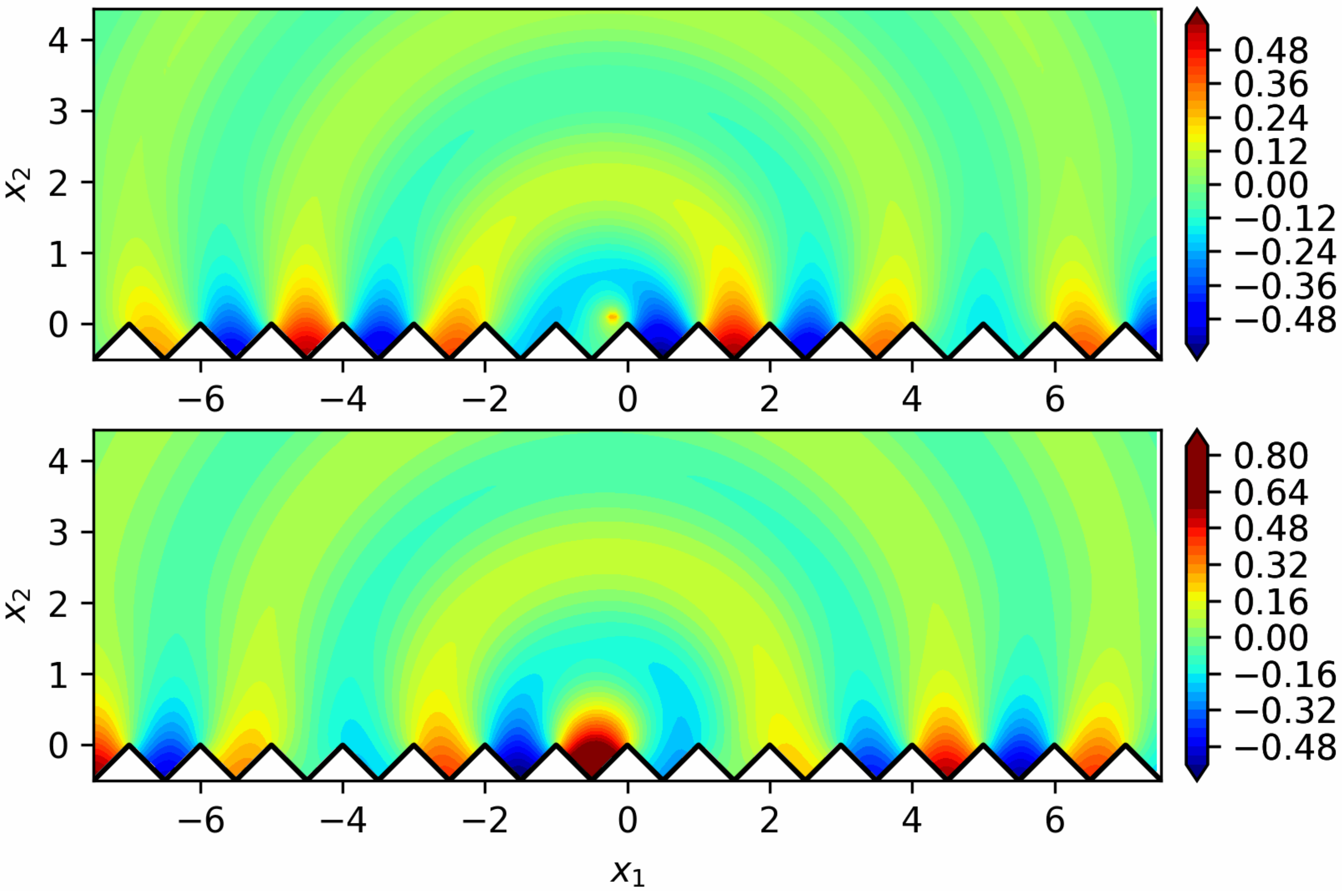
Array scanning / Floquet—Bloch transform

- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ

$$u(\mathbf{x}) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})\mathrm{d}\kappa,$$

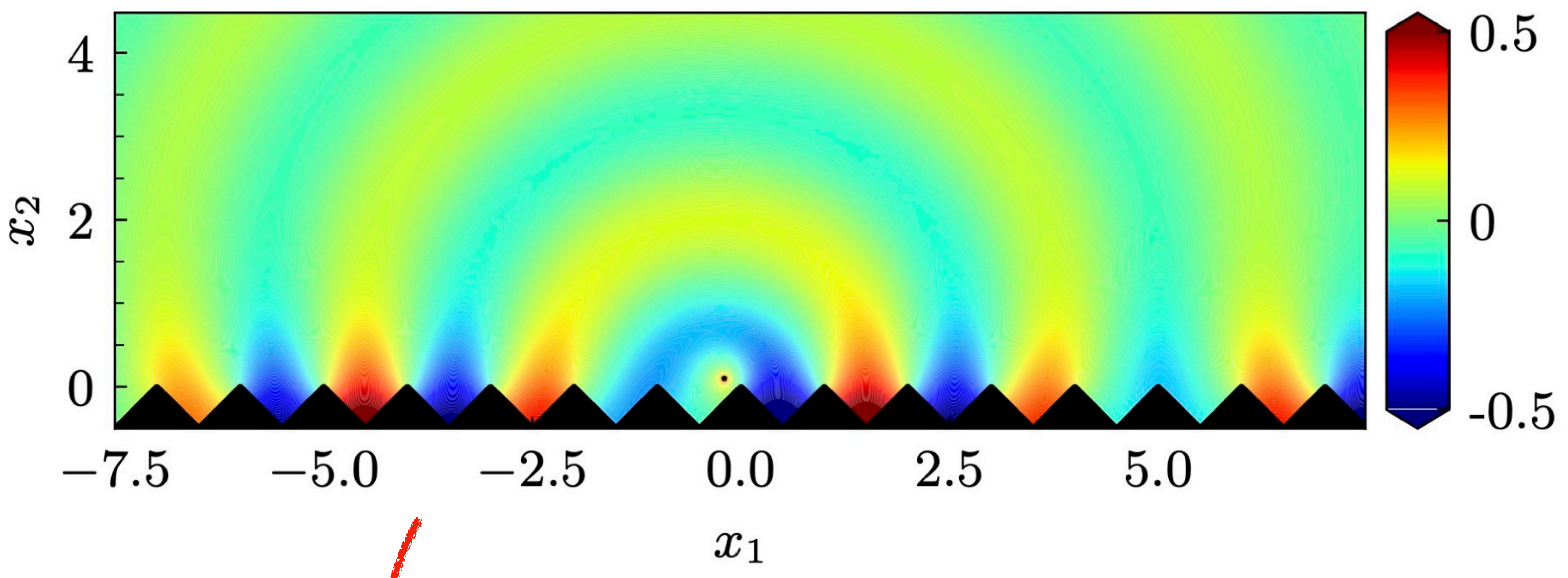
  $\rightarrow$ solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)

- But, on real axis:
  - Branch cuts at Wood anomalies κ_W, squareroot singularity
  - Poles at trapped modes κ_tr
- Contour deformation (example path shown), sinusoidal with amplitude A, trapezoidal rule with P_asm nodes*
- Direction of branch cuts/contour obeys the **limiting absorption principle:** $u(\mathbf{x}, t) = u(\mathbf{x})e^{-i\omega t}$ correspond to outgoing waves



Array scanning / Floquet—Bloch transform

- **A neat trick**: write point source as an integral of quasiperiodic sets of point sources over the horizontal wavenumber κ
$$u(\mathbf{x}) = \frac{d}{2\pi}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})\mathrm{d}\kappa,$$
→ solution for a single point source is quasiperiodic solution integrated over all possible wavenumbers $\kappa \in \left[-\frac{\pi}{d}, \frac{\pi}{d}\right]$. (*Munk & Burrell, IEEETAP, 1979*)
- But, on real axis:
    - Branch cuts at Wood anomalies κ_W, squareroot singularity
    - Poles at trapped modes κ_tr
- Contour deformation (example path shown), sinusoidal with amplitude A, trapezoidal rule with P_asm nodes*
- Direction of branch cuts/contour obeys the **limiting absorption principle:** $u(\mathbf{x}, t) = u(\mathbf{x})e^{-i\omega t}$ correspond to outgoing waves



Accurate to 8 digits

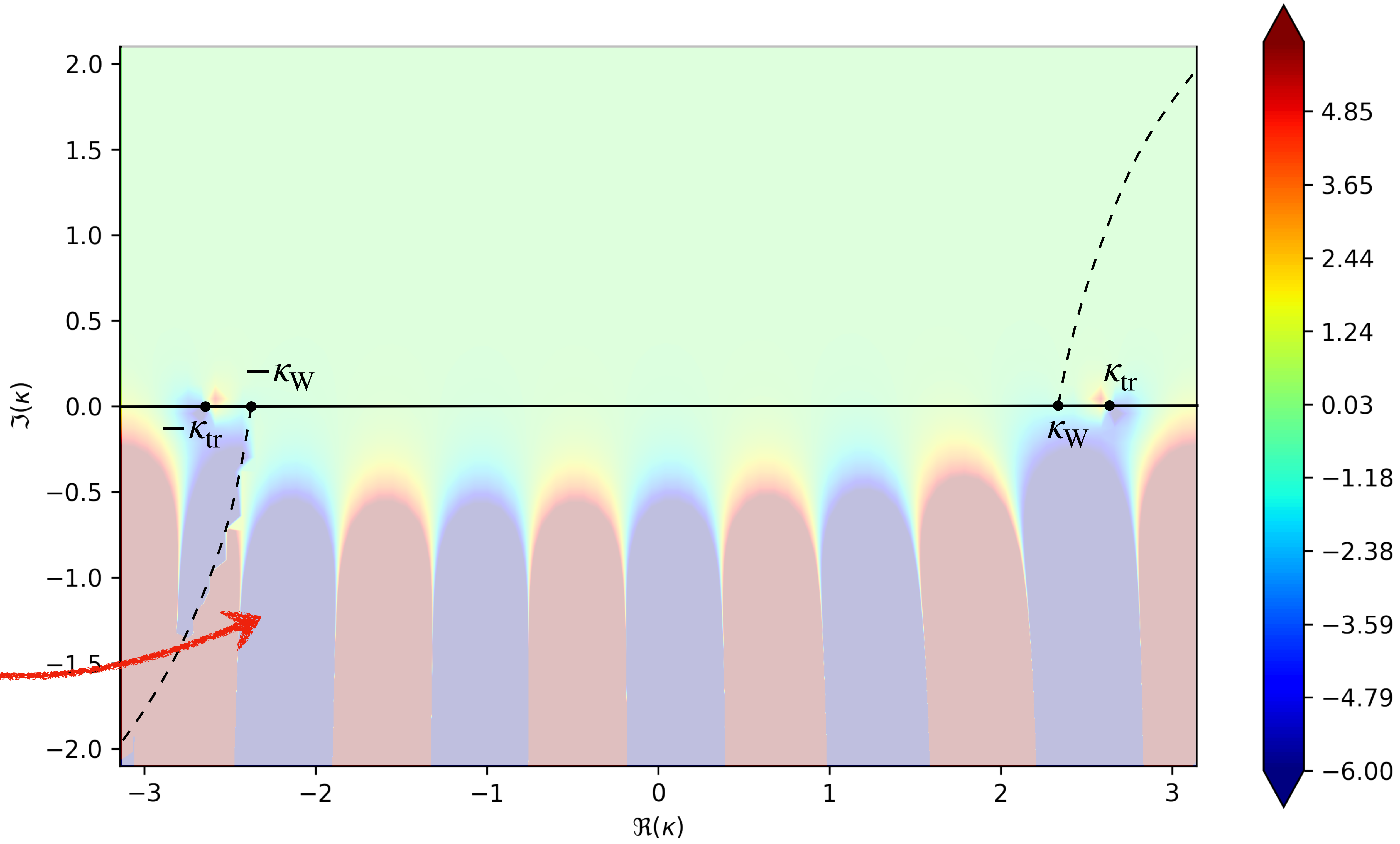
Power distribution in trapped modes

- What fraction of the total flux is transported in trapped modes? **Compute via residue of pole.**

- Claim: **in the far-away limit near the surface, only trapped mode remains,** i.e. only contribution to κ-integral will be from $\kappa = \kappa_{\text{tr}}$

- **Why?** Take solution in the limit of n (cell index) $\to \infty$,

$$\lim_{n\to+\infty} u(\mathbf{x}+n\mathbf{d}) = \frac{d}{2\pi} \lim_{n\to+\infty} \int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x}) e^{in\kappa} \mathrm{d}\kappa.$$



Power distribution in trapped modes

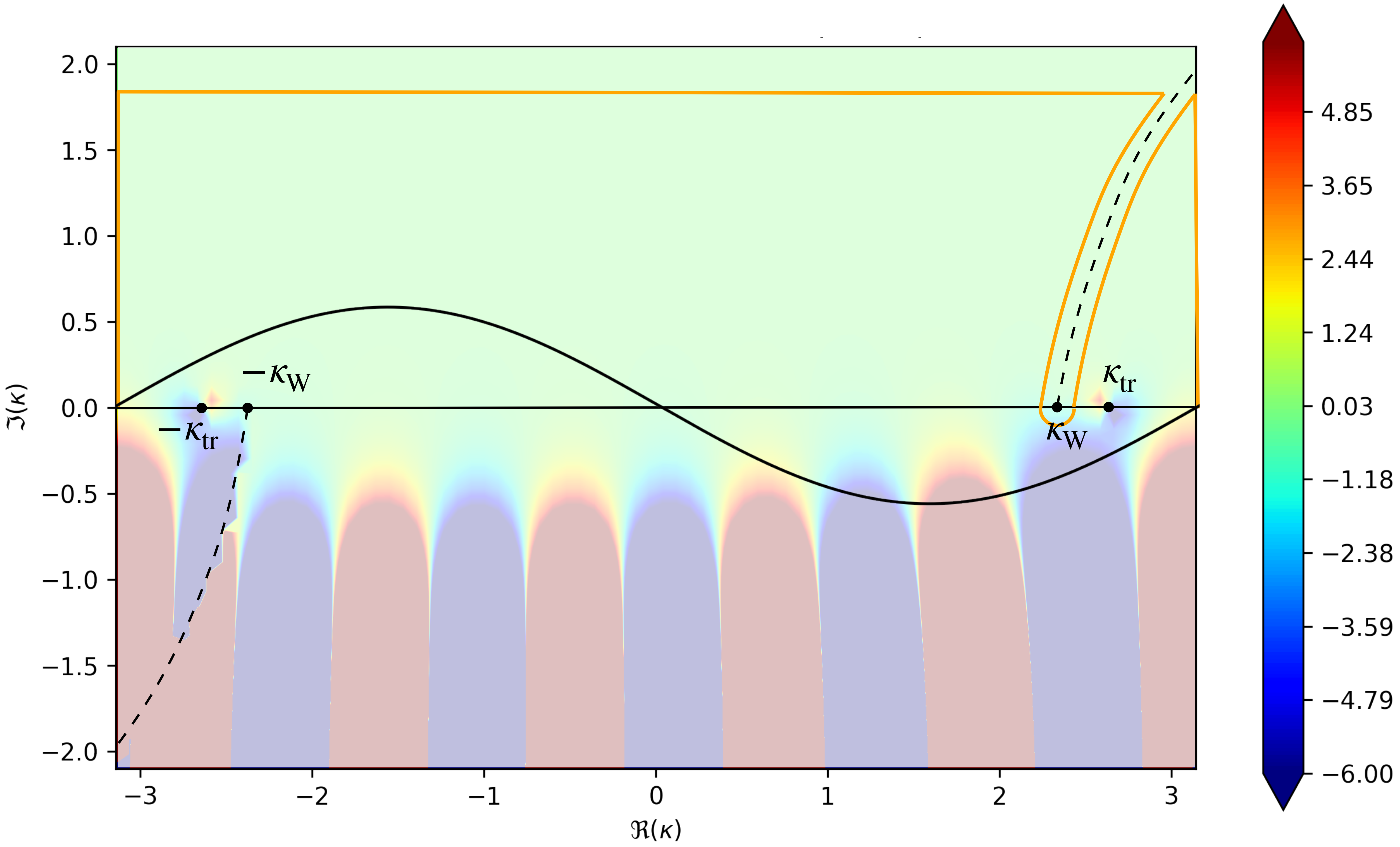
- What fraction of the total flux is transported in trapped modes? **Compute via residue of pole.**

- Claim: **in the far-away limit near the surface, only trapped mode remains,** i.e. only contribution to κ-integral will be from $\kappa = \kappa_{\text{tr}}$

- **Why?** Take solution in the limit of n (cell index) $\to \infty$,

$$\lim_{n \to +\infty} u(\mathbf{x} + n\mathbf{d}) = \frac{d}{2\pi} \lim_{n \to +\infty} \int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x}) e^{in\kappa} \mathrm{d}\kappa.$$

Close deformed contour in upper half plane $\to$ only residue of **right-hand pole** remains.



Power distribution in trapped modes

- What fraction of the total flux is transported in trapped modes? **Compute via residue of pole.**
- Claim: **in the far-away limit near the surface, only trapped mode remains,** i.e. only contribution to κ-integral will be from $\kappa = \kappa_{\text{tr}}$
- **Why?** Take solution in the limit of n (cell index) $\rightarrow \infty$,

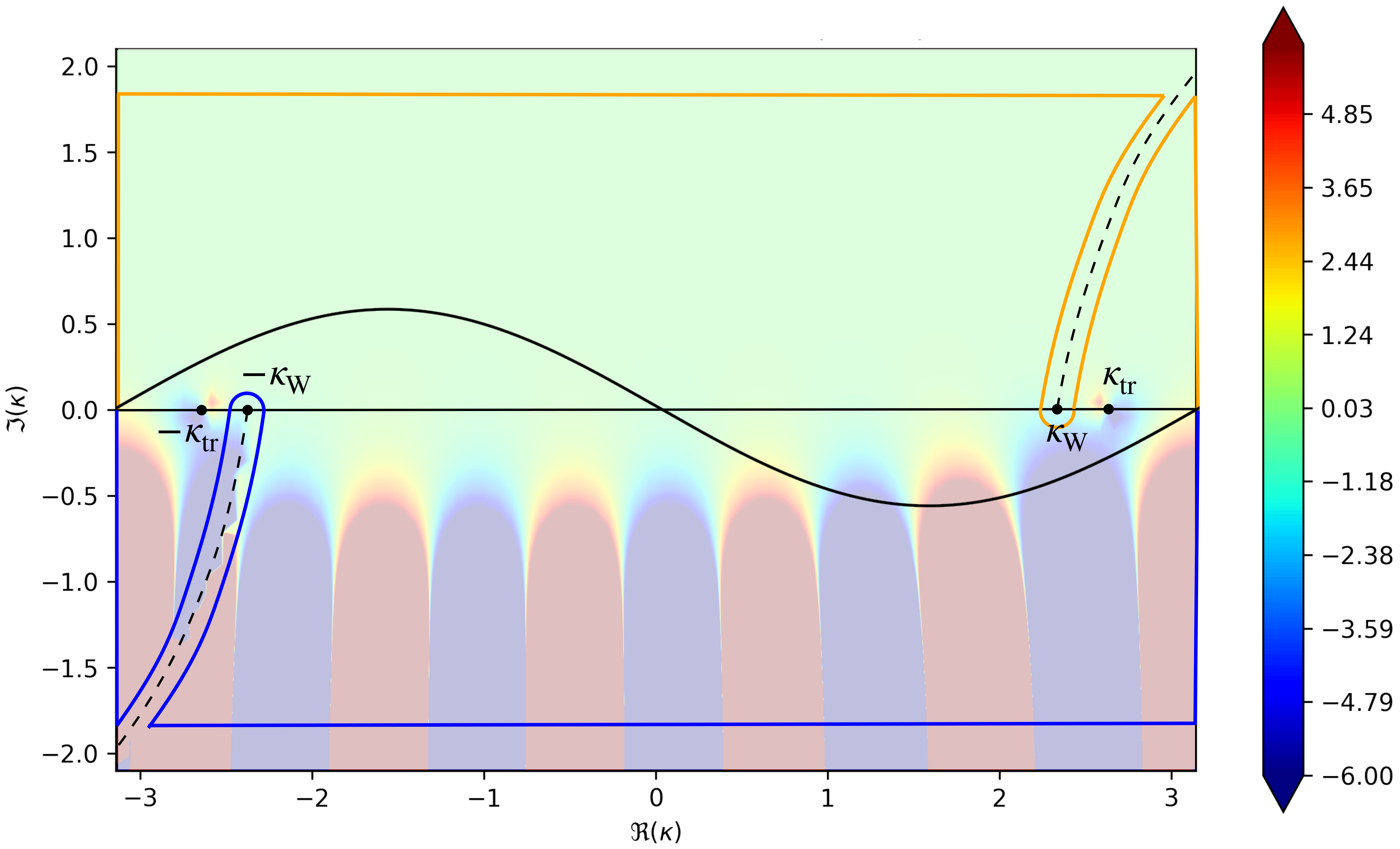
$$\lim_{n\to+\infty} u(\mathbf{x}+n\mathbf{d}) = \frac{d}{2\pi}\lim_{n\to+\infty}\int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x})e^{in\kappa}\mathrm{d}\kappa.$$

Close deformed contour in upper half plane $\rightarrow$ only residue of **right-hand pole** remains. Therefore,

$$\lim_{n\to+\infty} u(\mathbf{x}+n\mathbf{d}) = i\mathrm{Res}_{\kappa=\kappa_{\text{tr}}} u(\mathbf{x}) \quad \text{up to a complex phase.}$$

For $n \rightarrow -\infty$, **residue of left-hand pole dictates.**

- Compute residues numerically, on a small circle around κ_{tr} with trapezoidal rule



Power distribution in trapped modes

- What fraction of the total flux is transported in trapped modes? **Compute via residue of pole.**
- Claim: **in the far-away limit near the surface, only trapped mode remains,** i.e. only contribution to κ-integral will be from $\kappa = \kappa_{\text{tr}}$
- **Why?** Take solution in the limit of n (cell index) $\to \infty$,

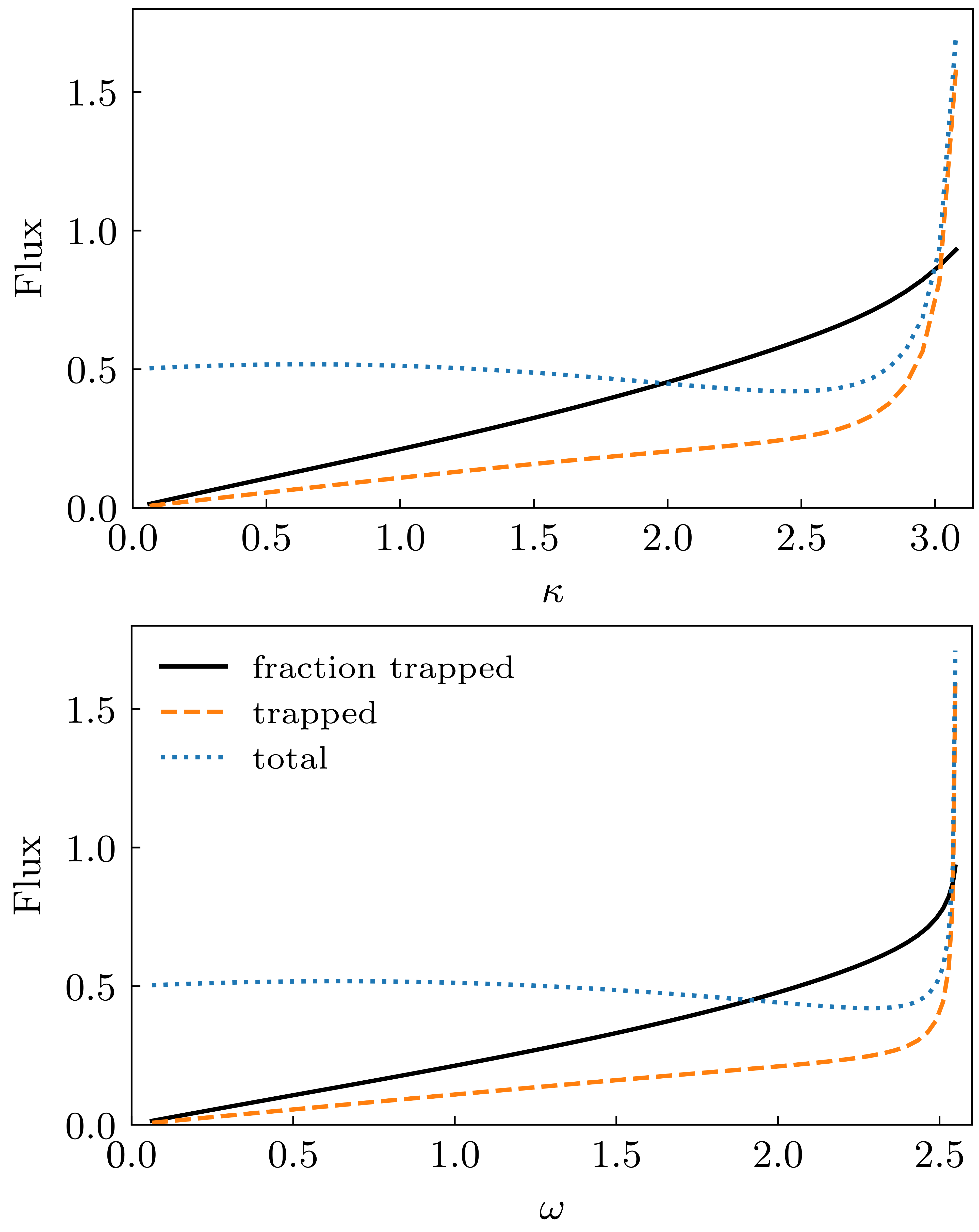
$$\lim_{n\to+\infty} u(\mathbf{x} + n\mathbf{d}) = \frac{d}{2\pi} \lim_{n\to+\infty} \int_{-\pi/d}^{\pi/d} u_\kappa(\mathbf{x}) e^{in\kappa} \mathrm{d}\kappa.$$

Close deformed contour in upper half plane $\to$ only residue of **right-hand pole** remains. Therefore,

$$\lim_{n\to+\infty} u(\mathbf{x} + n\mathbf{d}) = i\mathrm{Res}_{\kappa=\kappa_{\text{tr}}} u(\mathbf{x}) \quad \text{up to a complex phase.}$$

For $n \to -\infty$, **residue of left-hand pole dictates.**

- Compute residues numerically, on a small circle around κ_{tr} with trapezoidal rule



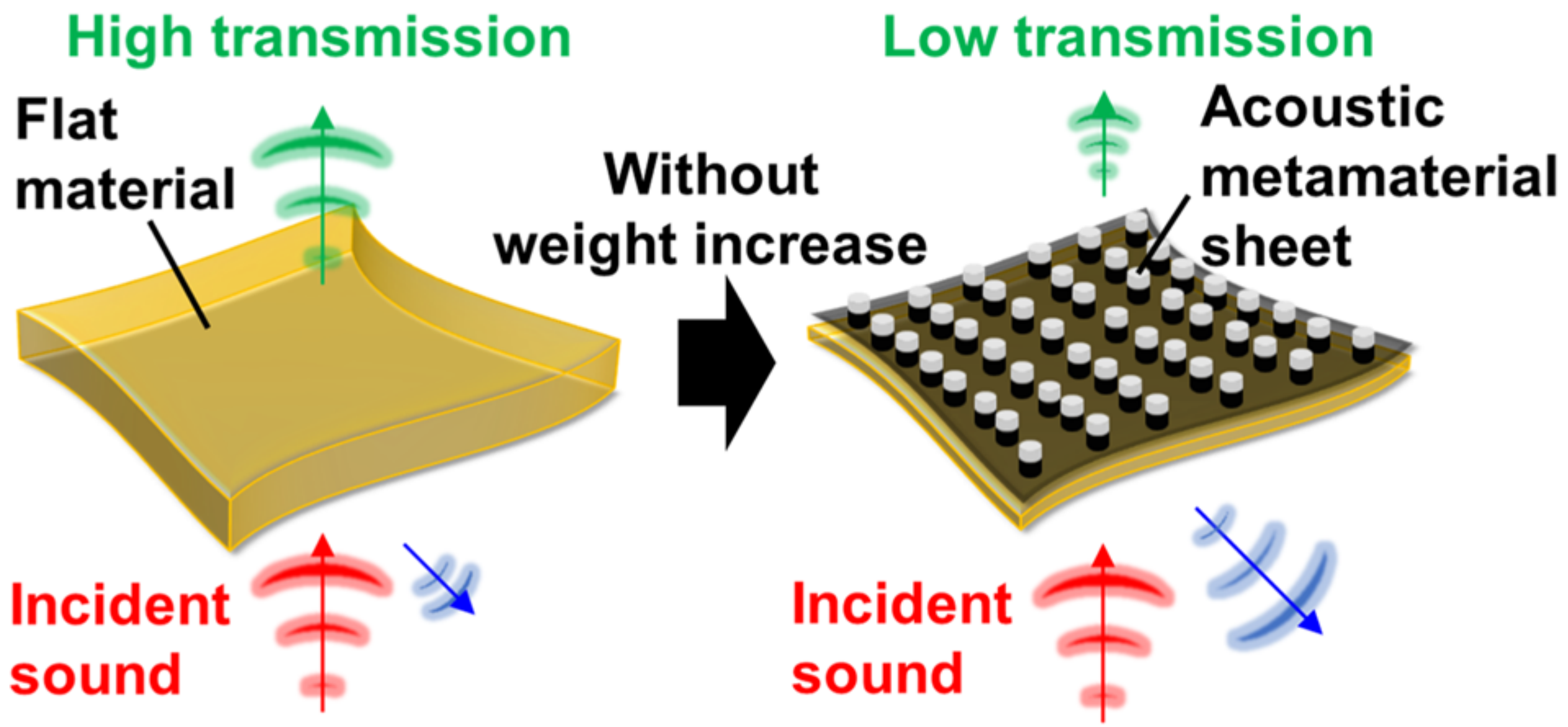
Future work & applications

- How does the *position of the source* and the *geometry* affect the power distribution in trapped modes?
  - Can a left/right flux asymmetry be induced?
  - What happens in asymmetric geometries?
  - Derive a fast, approximate model for the power distribution

Could make good PhD student projects

Future work & applications

- How does the *position of the source* and the *geometry* affect the power distribution in trapped modes?
  - Can a left/right flux asymmetry be induced?
  - What happens in asymmetric geometries?
  - Derive a fast, approximate model for the power distribution
- 3D generalization
  - Doubly periodic surfaces: **acoustic and seismic filtering, nondestructive testing of thin materials**
    - band structure complex, poles are lines
  - Triply periodic lattices
- Inverse problem for **fault detection** in periodic structures (e.g. photonic crystals)
- Klein—Gordon equation for **topological insulators**



From *Nakayama, Polymer Journal, 2024.* “Acoustic metamaterials based on polymer sheets: from material design to applications as sound insulators and vibration dampers”

Thank you

*Mountain bluebirds get their colors not from pigment but from diffraction by (microscopic) periodic structures in their feathers.

Credit: Mountain Bluebird by Krista Hinman; Cornell Lab of Ornithology | Macaulay Library