HarmonicBalance.jl: A Julia suite for nonlinear dynamics using harmonic balance

Jan Košata[1,*], Javier del Pino[1,*], Toni L. Heugel[1], and Oded Zilberberg[2]

1 Institute for Theoretical Physics, ETH Zürich, 8093 Zürich, Switzerland
2 Department of Physics, University of Konstanz, 78464 Konstanz, Germany
* J. K. and J. d. P. contributed equally to this work.
kosataj@phys.ethz.ch, jdelpino@phys.ethz.ch

February 2, 2022

Abstract

HarmonicBalance.jl is a publicly available Julia package designed to simplify and solve systems of periodic time-dependent nonlinear ordinary differential equations. Time dependence of the system parameters is treated with the harmonic balance method, which approximates the system's behaviour as a set of harmonic terms with slowly-varying amplitudes. Under this approximation, the set of all possible steady-state responses follows from the solution of a polynomial system. In HarmonicBalance.jl, we combine harmonic balance with contemporary implementations of symbolic algebra and the homotopy continuation method to numerically determine *all* steady-state solutions and their associated fluctuation dynamics. For the exploration of involved steady-state topologies, we provide a simple graphical user interface, allowing for arbitrary solution observables and phase diagrams. HarmonicBalance.jl is a free software available at https://github.com/NonlinearOscillations/HarmonicBalance.jl.

1 Introduction

Nonlinear ordinary differential equations (ODE) describe the time evolution of physical systems, in which distinct parts of the system interfere, cooperate or compete nonlinearly. Such ODEs host unique phenomena, including bifurcations, synchronisation, and chaos [1–3]. These manifest in a plethora of fields, such as fluid dynamics [4], mechanics and robotics [5], structural engineering [6], electronics [7], optics [8], predator-prey dynamics [9], chemical oscillations [10], and biological processes [11]. In particular, explicitly time-dependent or *non-autonomous* nonlinear ODEs describe driven-dissipative systems, which commonly exhibit harmonic time dependence of system parameters and/or external drives [12], see Fig. 1a.

Despite the ubiquitous presence of nonlinear dynamical systems in science and engineering [Fig. 1b], their behaviour is not analytically tractable in most cases. Hence, one often resorts to using numerical ODE solvers. These usually focus on *initial value problems*, where the system's state is advanced from a set of initial conditions. Linear systems, given sufficient time to freely evolve, usually relax to a unique *stationary* or *steady state*, i.e., a state where the system no longer evolves in time. In a harmonically driven system, the steady states will typically display time dynamics which are also harmonic, requiring a corresponding definition of a dynamical stationarity condition.

Nonlinear systems challenge this approach since they can feature multiple steady-state solutions. The knowledge of *all* such solutions is of key importance for analysing the system's behaviour, revealing experimentally observable phenomena such as hysteresis, spontaneous symmetry-breaking, or noise-induced switching dynamics [13–15]. However, when using a dynamical solver (or a

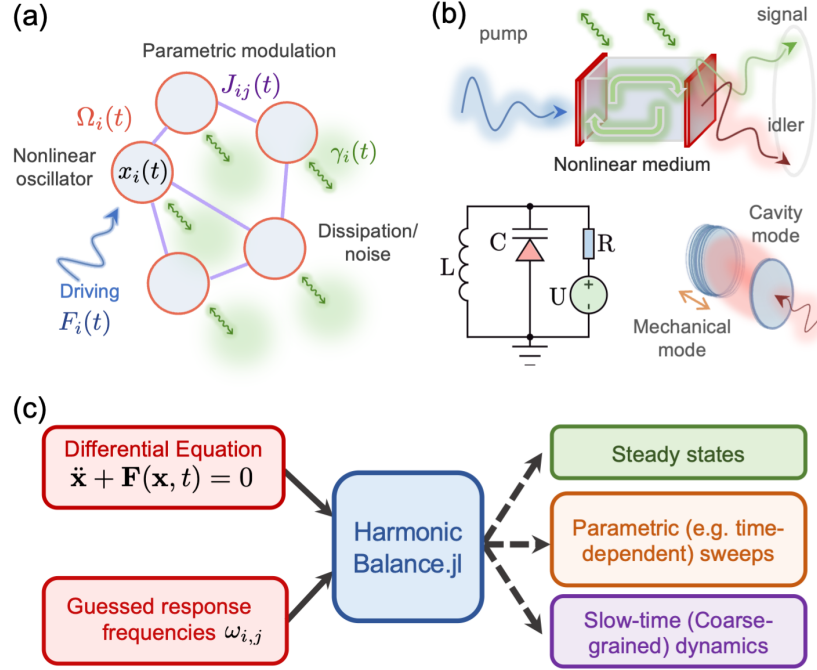



Figure 1: **Example systems treatable by HarmonicBalance.jl.** (a) A generic driven dissipative system: coupled nonlinear oscillators (circles) with amplitudes $x_i(t)$, time-varying natural frequencies $\Omega_i(t)$, and coupling amplitudes $J_{ij}(t)$ (here $i, j = 1, 2, \cdots, N$). System-environment interactions, parameterised by $\gamma_i(t)$, lead to fluctuations and dissipation. Harmonic drives with amplitudes $F_i(t)$ excite and modify the system's response. (b) Examples of nonlinear devices found in physics and engineering: (top) light fields interacting with nonlinear media, (bottom, left) driven nonlinear electric RLC circuits, and (bottom right) optomechanical resonators. (c) Basic functionality of HarmonicBalance.jl: A nonlinear (polynomial) ODE system with harmonic time-dependence is defined along with a set of expected response frequencies for each resonator, ω_{ij} (red input boxes). The package finds all dynamical steady states, identifies the most likely response given a parametric sweep, and supports numerical time-evolution within the harmonic ansatz (introduced in 2.1).

time-resolved experiment), only a single steady state is found per run, depending on the initial conditions [16, 17]. Therefore, a complete exploration of the solution landscape would require infinite sampling of the continuous space of initial conditions.

An alternative approach to finding steady-state solutions in harmonic systems is transforming the system into a frame rotating with the applied drives [18, 19]. In such a frame, the drives and corresponding steady states appear stationary, reducing the problem to finding the roots of a time-independent system of nonlinear equations [20, 21]. While this approach is standard in few-variable problems, the proliferation of roots in multivariate nonlinear systems constitutes a challenge in itself [22–24]. Indeed, the most straightforward root-finding algorithms (e.g., the Newton-Raphson method) only find a single solution in the vicinity of an initial guess, making the exploration of all steady states infeasible.

Fortunately, many of the aforementioned nonlinear examples [cf. Fig. 1b] feature ODE systems with polynomial dependence on the dynamical variables and their derivatives. For these, seeking

all steady states in a given rotating frame generates coupled polynomial (algebraic) equations, which are numerically solvable using the *homotopy continuation* method [25–27]. This method utilises the continuous deformation of an exactly solvable problem into the problem of interest, thus finding *all* the roots of coupled polynomials in one go. To our knowledge, however, no homotopy-continuation-based toolbox for the analysis of steady states of harmonically-driven systems has been developed so far.

Here, we introduce HarmonicBalance.jl: an open-source suite that can find steady-state solutions to non-autonomous differential equations with harmonic time-dependencies. We unify a variety of existing methods for the analysis of time-dependent nonlinear ODEs into an integrated framework and take advantage of open-source Julia libraries to achieve ease of use, flexibility, and high performance [Fig. 1c]. Our package is readily applicable to a range of active fields where nonlinear harmonically-driven systems appear, such as modal analysis in structural dynamics [6,28], electric circuits [7,29,30], nonlinear optics [8,31–37], optomechanics [17,38–40], micro- and nanomechanics [41–50], oscillator networks [51–56], Ising machines [57–62], and many-body light-matter systems [63–69].

2 Harmonically-driven nonlinear systems: basic principles

HarmonicBalance.jl focuses on harmonically-driven nonlinear systems, i.e., dynamical systems governed by equations of motion where all explicitly time-dependent terms are harmonic. In the exposition here, we will assume a general nonlinear system of N second-order ODEs[1] with real variables $x_i(t)$, with index $i = 1, 2, \cdots, N$ and time t as the independent variable,

$$\ddot{\mathbf{x}}(t) + \mathbf{F}(\mathbf{x}(t), t) = 0. \tag{1}$$

The vector $\mathbf{x}(t) = (x_1(t), ..., x_N(t))^{\mathrm{T}}$ fully describes the state of the system. Physically, $\mathbf{x}(t)$ encompasses the amplitudes of either point-like or collective oscillators (e.g., mechanical resonators, voltage oscillations in RLC circuits, an oscillating electrical dipole moment, or standing modes of an optical cavity). We assume $\mathbf{F}(\mathbf{x}(t), t)$ can be decomposed into a sum of L periodic terms[2]:,

$$\mathbf{F}(\mathbf{x}, t) = \mathbf{f}_0(\mathbf{x}) + \sum_{l=1}^{L} \mathbf{f}_l(\mathbf{x}) \cos(\omega_l t + \phi_l), \tag{2}$$

with vector fields $\mathbf{f}_l(\mathbf{x})$. The field $\mathbf{f}_0$ describes static properties of the system while $\mathbf{f}_{l \neq 0}$ account for explicit time-dependence, typically induced by one or more sources of periodic driving and/or parameter modulation with frequencies $\{\omega_l\}$ and phases $\{\phi_l\}$. In Table 1, we list several examples of terms commonly constituting Eq. (2).

Note that in a linear system, $\mathbf{F}(\mathbf{x}, t)$ can be written as $\mathbf{F}(\mathbf{x}, t) = \mathbf{M}(t)\mathbf{x} + \mathbf{b}(t)$, where the matrix $\mathbf{M}(t)$ contains spring constants and linear couplings, while $\mathbf{b}(t)$ is a vector of external forces. Diagonalising $\mathbf{M}(t)$ then yields the so-called *normal modes* of the system. While the notion of normal modes does not directly apply in a nonlinear system, they usually constitute a convenient basis choice for a perturbative treatment.

---

[1]Second-order ODEs based on the harmonic oscillator represent the vast majority of expected use cases. However, the methods described in this work are applicable to arbitrary ODE orders [12].

[2]We omit the time-dependence of $x(t)$ where this is clear from the context.

term in $F_i(\mathbf{x}, t)$	physical mechanism	frequency conversion
x_i	natural response (spring constant)	-
x_j	mode coupling	-
$\dot{x}_i$	damping/gain	-
$\dot{x}_j$	dissipative coupling	-
$\cos(\omega_d t)$	external drive (frequency ω_d)	-
x_i^2	Pockels coefficient	$\omega_i \to 2\omega_i$
x_i^3	Kerr (Duffing) coefficient	$\omega_i \to 3\omega_i$
$x_i^2 \dot{x}_i$	nonlinear damping	$\omega_i \to 3\omega_i$
$x_i \cos(\omega_p t)$	parametric drive (frequency ω_p)	$\omega_i \to \omega_i \pm \omega_p$
$x_j x_i$	nonlinear mode interactions	$\omega_i \to \omega_i \pm \omega_j$

Table 1: **Examples of terms occurring in Eq.** (2) **and their origins.** The rightmost column shows the frequency conversion taking place, assuming two variables x_i, x_j, oscillating at frequencies ω_i, ω_j (for example, $\omega_i \to 2\omega_i$ means, that oscillating at frequency ω_i leads to additional oscillations at frequency $2\omega_i$).

2.1 The harmonic expansion

For sufficiently long times (i.e., after any transient responses have disappeared), the solutions of Eq. (1) are expected to appear as a sum over harmonics. Let us illustrate this point using the example of driven harmonic oscillators. In this simple case, Eqs. (1) and (2) take the form

$$\ddot{\mathbf{x}}(t) + \mathbf{M}\mathbf{x}(t) = \mathbf{g}\cos(\omega_d t), \tag{3}$$

where we assumed a constant vector $\mathbf{g}$ and drive frequency ω_d. The standard method to solve for the steady states is to Fourier-transform both sides of Eq. (3); the resulting equations give an exact solution for $\mathbf{x}(t)$ in terms of its Fourier coefficients,

$$\tilde{\mathbf{x}}(\omega) = (\mathbf{M} - \omega^2 \mathbb{1})^{-1} \mathbf{g} [\delta(\omega + \omega_d) + \delta(\omega - \omega_d)]/2, \tag{4}$$

where $\tilde{\mathbf{x}}(\omega) = \int_{-\infty}^{+\infty} \mathbf{x}(t)e^{i\omega t}\, dt$ and $\delta(z - z_0)$ is the Dirac delta function of z centred at z_0. This procedure is effective because in the Fourier domain, the l.h.s. of Eq. (3) becomes diagonal (i.e., it involves a single frequency) while the applied drive reduces to a Dirac delta function centred at $\omega = \omega_d$. We can thus select a single frequency out of the continuous space of all frequencies for which the solution is nonvanishing. Furthermore, due to the linear superposition principle, responses to arbitrary driving terms can be constructed out of the solution in Eq. (4).

The same task becomes intractable if nonlinear terms are introduced. Nonlinearities facilitate *frequency conversion* by coupling different harmonics of the system, rendering $\mathbf{F}(\mathbf{x}, t)$ non-diagonal in Fourier space. As an example, let us consider a single driven Duffing oscillator [70], whose equation of motion reads

$$\ddot{x}(t) + \omega_0^2 x(t) + \alpha x^3(t) = F\cos(\omega_d t). \tag{5}$$

where ω_0 is the natural frequency, α is the nonlinear coefficient, F is the drive amplitude, and x is now a scalar. The nonlinear (Duffing) term in Fourier space reads

$$\alpha \int x^3(t)e^{-i\omega t}\, dt = \alpha \int_{-\infty}^{+\infty} \tilde{x}(\omega')\tilde{x}(\omega'')\tilde{x}(\omega''')\delta(\omega''' + \omega'' + \omega' - \omega)\, d\omega'\, d\omega''\, d\omega''', \tag{6}$$

coupling all combinations of four harmonics that sum to zero. This results in frequency-conversion processes known as four-wave mixing, which here convert, to lowest order, the driven oscillation at frequency ω_d to frequency $3\omega_d$; four-wave mixing appears as off-diagonal terms in Fourier space. The frequency conversion subsequently propagates through the entire spectrum, generating an infinite number of Fourier components. A nonlinearity thus precludes a closed-form solution of the problem in Fourier space; this is a common trait of nonlinear physical systems; see Table 1 for examples of frequency-converting effects.

A serviceable approach to numerically approximate the harmonics of a driven nonlinear system involves truncating the spectrum $\tilde{\mathbf{x}}(\omega)$ to a finite set of frequencies [71]. The idea of truncation in Fourier space is at the core of several widely-used methods, such as harmonic balance [72, 73], the rotating-wave approximation [74], the van der Pol transformation [3] in combination with Krylov-Bogoliubov averaging [75], Magnus expansion [76, 77], secular perturbation theory [70], and also appears in the contemporary concept of Floquet engineering [78, 79]. We implement the approach of harmonic balance using a generalised ansatz [80, 81]

$$x_i(t) = \sum_{j=1}^{M} u_{i,j}(T)\cos(\omega_{i,j}t) + v_{i,j}(T)\sin(\omega_{i,j}t), \tag{7}$$

where the sum runs over the finite set of desired frequencies $\{\omega_{i,j}\}$ describing the coordinate x_i. Here, T represents a "coarse-grained" timescale that is much slower than the oscillations in the system ($T \gg 2\pi/\min\{\omega_j\}$). Equation (7) represents an attempt to capture the dynamics of the system using a discrete set of real functions $\{u_{i,j}(T), v_{i,j}(T)\}$, which in our package we dub the *harmonic variables*, while the fast oscillations are accounted for by the sine/cosine terms. Note that this ansatz would be exact if the set of frequencies $\{\omega_{i,j}\}$ covered all real numbers. Generally, it is not straightforward to identify the relevant frequencies $\{\omega_{i,j}\}$. A good starting point is the frequencies of any external drives, combined with the frequency conversions presented in Table 1. Alternatively, one can pick the highest-weight discrete Fourier components in a time trace obtained numerically or from experimental data.

Plugging the ansatz Eq. (7) into the ODE (1), we obtain equations governing pairs of harmonic variables $(u_{i,j}, v_{i,j})$ by convolving both sides with $\cos(\omega_{i,j}t)$ and $\sin(\omega_{i,j}t)$, respectively (see Appendix A). The harmonic variables themselves are treated as constants during this step. We thus obtain two equations for each $\omega_{i,j}$. To reflect the slowly-changing nature of the harmonic variables, we drop all of their time derivatives of order two or higher, e.g., $\ddot{u}_{i,j}, \ddot{v}_{i,j} \to 0$. The resulting set of equations has no explicit time dependence; we call these the *harmonic equations*,

$$\frac{d\mathbf{u}(T)}{dT} = \bar{\mathbf{F}}(\mathbf{u}), \tag{8}$$

where $\mathbf{u}(T) = (u_{1,1}(T), v_{1,1}(T), ..., u_{N,M}(T), v_{N,M}(T))^{\mathrm{T}}$ contains the harmonic variables and $\bar{\mathbf{F}}(\mathbf{u})$ the corresponding Fourier components resulting from inserting the truncated ansatz Eq. (7) into Eq. (2). Note that Eq. (8) still describes a slow time dependence of $\{u_{i,j}(T), v_{i,j}(T)\}$, or in other words, Eq. (8) captures an intrinsic frequency bandwidth $\Delta\omega \ll \omega_{i,j}$ of the response around the expansion frequencies $\{\omega_{i,j}\}$ in a similar spirit as the Krylov-Bogoliubov averaging [13, 75] and the rotating-wave-approximation [74]. Note that our scheme scales up very fast - each harmonic $\omega_{i,j}$ of a variable x_i is converted into two harmonic variables. Hence, for a system of N interacting components, each expanded in M harmonics, Eq. (8) consists of $2NM$ harmonic equations.

2.2 Solving the harmonic equations for steady-state solutions

There are, in principle, two ways to extract the long-time behaviour of the system from Eq. (8). First, one can make use of an initial-value ODE solver to numerically propagate a state in T, starting at $\mathbf{u}(T = T_0)$. Given sufficient time, the system will typically converge towards a steady state $\mathbf{u}_0$. Secondly, we can look for steady states by explicitly requiring the l.h.s. of Eq. (8) to vanish, leaving us with the task of solving the nonlinear *algebraic* equations system

$$\bar{\mathbf{F}}(\mathbf{u}_0) = 0 \,. \tag{9}$$

The second approach is advantageous since, in principle, all steady states of the system can be found, irrespective of whether or how they can be reached by time evolution. In general, however, obtaining the roots of coupled nonlinear algebraic equations is highly non-trivial: for n coupled polynomial equations of order p, Bézout's theorem [82] provides an upper bound p^n for the number of distinct complex solutions. For example, in the Duffing oscillator shown in Eq. (5), we have one variable x - given an ansatz Eq. (7) with a single harmonic. This generates 2 polynomial harmonic equations of order 3, leading to a maximum of $3^2 = 9$ solutions. Generally, for N coupled Duffing oscillators each expanded in M harmonics, we have $2NM$ harmonic equations of order 3, leading to an upper bound of 3^{2NM} solutions. The exponential scaling rapidly makes charting the complete steady-state solution landscape a formidable challenge. Accordingly, many applications of harmonic balance deal with few-variable systems and rely on the perturbative treatment of nonlinearities.

In our package, we solve the algebraic Eq. (9) using the homotopy continuation method [83,84] as implemented by the open-source package HomotopyContinuation.jl [26]. To find the roots of a polynomial, this method, in its simplest form, starts from another analytically-solvable polynomial of the same order, which saturates the Bézout bound. For instance, to find the roots of a single, p-th order polynomial $P(z)$ of the variable z, one can start from the polynomial $U(z) = z^p - 1$ with roots $e^{2\pi i k/p}$ ($k = 1, ..., p$). Employing this so-called *total-degree homotopy*, the polynomial $U(z)$ is "slowly" deformed [3] into the polynomial $P(z)$, correcting the known roots (i.e., tracking the roots) after each step. At the end of the homotopy continuation, the obtained set of roots is guaranteed to be complete, as it has been tracked from the full set of p complex roots of $U(z)$.

A generalisation of the total degree homotopy approach enables the solution of Eq. (9), by instead tracking the roots of the uncoupled system $\mathbf{U}(\mathbf{u}_0) = \{u_1^{d_1} - 1, v_1^{d_2} - 1, u_2^{d_3} - 1, v_2^{d_4} - 1, \cdots\}$, with d_r equal to the degree of the polynomial $\bar{F}_r(\mathbf{u}_0)$, to those of $\bar{\mathbf{F}}(\mathbf{u}_0)$ (here $r = 1, 2, \cdots, 2NM$ is an index labelling all harmonic balance equations). Such an approach leads to a number of solution paths to track that scales exponentially with NM. This constitutes a challenge in usual exploratory research scenarios that seek to find the behaviour of the solutions as parameters vary. Crucially, in physical problems (and in contrast with single polynomials), $\bar{\mathbf{F}}(\mathbf{u}_0)$ often displays a vastly smaller number of roots than the Bézout bound (see Examples). This implies that many solution paths from $\mathbf{U}(\mathbf{u}_0) = 0$ are usually irrelevant (singular[4]) since they do not lead to roots of $\bar{\mathbf{F}}(\mathbf{u}_0)$. In HarmonicBalance.jl, we harness a common strategy to reduce the computational overhead in multi-parameter root finding [25]. This strategy consists of the two-step algorithm described in Appendix B and exemplified in Fig. 2.

---

[3]Namely, via the homotopy $H(U(z), P(z), \lambda) = \lambda U(z) + (1 - \lambda) P(z)$, with deformation parameter λ. In the current context, the homotopy is a mapping such that $H(U(z), P(z), \lambda = 1) = U(z)$ and $H(U(z), P(z), \lambda = 0) = P(z)$.

[4]A solution $\mathbf{u}_0$ is denoted singular if the Jacobian matrix for $\bar{\mathbf{F}}(\mathbf{u})$ at $\mathbf{u} = \mathbf{u}_0$ has full rank. These solutions are associated with the intersection, reversal or divergences of solution paths, and their early detection is crucial in efficient path tracking.

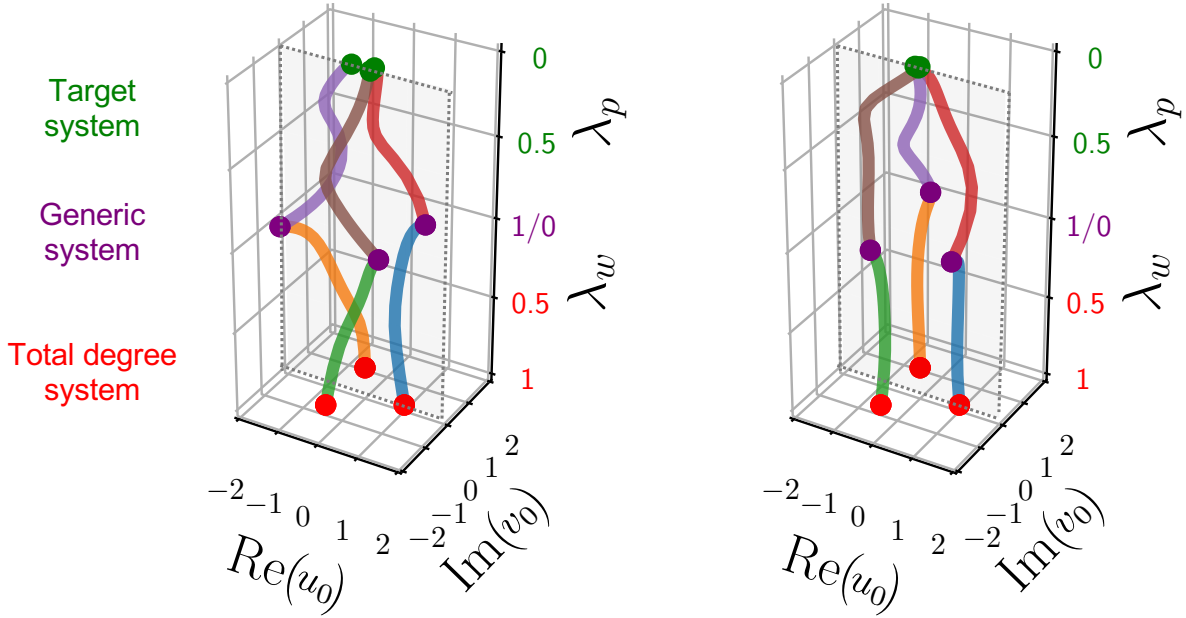



Figure 2: **Homotopy continuation in a Duffing resonator [cf. Eq. (5)]** Path tracking from the complex roots of the system $\{u_0^3 - 1 = 0, v_0^3 - 1 = 0\}$ (red dots) to the roots of the harmonic balance equations [cf. Eq. (9), Appendix A] (green dots), denoted $\mathbf{u}_0 = (u_0, v_0)$, using homotopy continuation. In the algorithm to track multiple parameter steady states, we carry out a two-step process encompassing a warm-up tracking (deformation parameter $\lambda_w \in (0, 1)$), where singular paths (not-shown) are filtered-out, followed by a subsequent tracking to the harmonic balance equations, (deformation parameter $\lambda_p \in (0, 1)$) [see Appendix B for further details]. The roots $\mathbf{u}_0$ traverse the complex plane until paths intersect with the planes $\text{Im}(u_0) = 0$ and $\text{Im}(v_0) = 0$ (light gray), at the 3 real roots of $\mathbf{F}(\mathbf{u}_0)$. For this figure $\alpha = 1$, $\omega_d = 1.03$, $\omega_0 = 1$, $F = 0.01$, $\theta = 0$, and $\gamma = 0.01$.

Finally, note that although the obtained solutions are complex, only real roots of Eq.(9) are physically meaningful[5]. For a more detailed description of the method and its implementation, see the documentation of HomotopyContinuation.jl and references therein [25, 26].

2.3 Stability analysis and linear response

Let us assume that we found a real solution $\mathbf{u}_0$ of Eq. (8). When the system is in this state, it responds to small perturbations either by returning to $\mathbf{u}_0$ over some characteristic timescale (*stable state*) or by evolving away from $\mathbf{u}_0$ (*unstable state*). To analyze the stability of $\mathbf{u}_0$, we linearize Eq. (8) around $\mathbf{u}_0$ for a small perturbation $\delta \mathbf{u} = \mathbf{u} - \mathbf{u}_0$ to obtain

$$\frac{d}{dT}[\delta \mathbf{u}(T)] = J(\mathbf{u} = \mathbf{u}_0)\delta \mathbf{u}(T), \tag{10}$$

where $J(\mathbf{u}_0) = \nabla_{\mathbf{u}}\bar{\mathbf{F}}|_{\mathbf{u}=\mathbf{u}_0}$ is the *Jacobian matrix* of the system evaluated at $\mathbf{u} = \mathbf{u}_0$[6].

---

[5]Complex roots with small imaginary parts suggest that small perturbations in the problem might introduce a new real root, see Ref. [25].

[6]In this notation, $\nabla_{\mathbf{u}}\bar{\mathbf{F}} = (\nabla_{\mathbf{u}}^{\mathrm{T}}\bar{F}_1, \nabla_{\mathbf{u}}^{\mathrm{T}}\bar{F}_2, \cdots, \nabla_{\mathbf{u}}^{\mathrm{T}}\bar{F}_{2NM})^{\mathrm{T}}$, where $\nabla_{\mathbf{u}}^{\mathrm{T}}\bar{F}_r$ is the transpose (row vector) of the gradient of the r-th component over the coordinates $\mathbf{u}$.

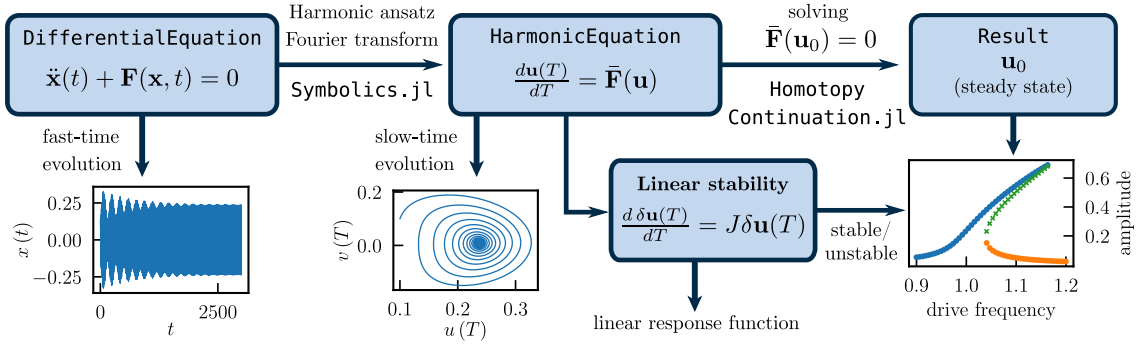



Figure 3: The basic workflow of HarmonicBalance.jl.

The solution to Eq. (10) can be expanded in terms of the complex eigenvalues λ_r and eigenvectors $\mathbf{v}_r$ of $J(\mathbf{u}_0)$, namely

$$\delta\mathbf{u}(T) = \sum_{r=1}^{2NM} (\mathbf{v}_r \cdot \delta\mathbf{u}(T = T_0))\, \mathbf{v}_r e^{\lambda_r T}. \tag{11}$$

The dynamical behaviour near the steady states is thus governed by $e^{\lambda_r T}$: if $\mathrm{Re}(\lambda_r) < 0$ for all r, the system returns to $\mathbf{u}_0$ under a small arbitrary perturbation and the state is classified as stable. Conversely, if $\mathrm{Re}(\lambda_r) > 0$ for any r, the system is unstable - perturbations such as noise or a small applied drive will force it to evolve away from $\mathbf{u}_0$.

The linear response of a stable, steady state to an additional oscillatory force, caused by weak probes or noise, is often observed in experiments [85]. It can be calculated by adding a small driving term $\delta\mathbf{f}\cos(\Omega_d T)$ to the harmonic Eq. (8). To account for the time dependence of the perturbation, linearisation around $\mathbf{u}_0$ should retain the previously-dropped higher-order time derivatives. While implemented in the package, we leave a thorough discussion of this topic to future work.

3 Structure of HarmonicBalance.jl

The bulk of HarmonicBalance.jl is written in Julia, a language combining the accessibility of interpreted languages such as Python with the performance of compiled languages such as C and Fortran. Our package is designed with simplicity of use and scalability in mind. We give a short overview of the package structure and basic workflow below; it is also displayed in Fig. 3.

1. ODE systems serve as primary input and are inserted in symbolic form via Symbolics.jl. Non-autonomous harmonic systems are accompanied by a user-defined set of frequencies to build the ansatz [cf., Eq.(7) and Fig. 1c]. From this, a set of harmonic (autonomous) equations is obtained symbolically.

2. Harmonic equations are interfaced with a root-finding algorithm (HomotopyContinuation.jl) and numerical ODE solvers (DifferentialEquations.jl).

3. Steady-state solutions are further analysed, e.g., for stability and other criteria. A flexible toolbox for the calculation of observable quantities is included, with a user-defined mapping of steady-state solutions and the calculation of linear response spectra.

4. Steady-state solutions for one- and two-dimensional parameter sets are readily visualised. Interactive plotting routines help exploratory analysis of the rich topology of the solutions (e.g., bifurcations) for multi-dimensional parameter sets. An ODE solver may be used to obtain the slow-time (T) dynamics and verify the steady-state results.

In the following, we detail the working principles of HarmonicBalance.jl. For detailed online documentation on the latest release, we refer the reader to [86].

3.1 Defining a system, extracting harmonic equations.

Conceptually, specifying a system requires two ingredients: its ODE of motion, such as Eq. (1), and the set of oscillating frequencies forming the harmonic ansatz in Eq. (7). Once defined, the equation of motion is stored in the dedicated object `DifferentialEquation`. For this primary input and subsequent symbolic manipulations, we employ the Julia package Symbolics.jl [87], whose emphasis on high performance is essential in dealing with complex problems, such as that shown in Section 5.3.

After constructing the harmonic ansatz Eq. (7) and introducing the slow time T, the harmonic equations governing the harmonic variables **u** are found. We assume slow or no evolution in T and therefore drop any second and higher-order derivatives with respect to T as well as products of $\frac{d\mathbf{u}}{dT}$. The equations are then Fourier-transformed, returning the harmonic Eq. (8) (for a concrete example, see Appendix A and Section 5.1). These equations are stored in the object `HarmonicEquation`, which in itself stores instances of `HarmonicVariable`, each specifying one pair $\{u_{i,j}, v_{i,j}\}$.

3.2 Obtaining and characterising steady states

To find and analyse the steady states of a `HarmonicEquation`, we need the corresponding (algebraic) steady-state equations and the Jacobian matrix; these are symbolically stored in the object `Problem`.

As the next step, numerical parameter values are specified. This is the required input for our primary solving method, `get_steady_states`, which allows the user to specify which parameters are constant and which are varied. One may vary any number of parameters, i.e., solution sets of any dimension are supported. We then employ a homotopy to retrieve steady states for each parameter set value. The native multi-threading support of HomotopyContinuation.jl can dispatch path tracking over multiple cores. Note that complex roots are being followed throughout the procedure - real solutions are only filtered out a posteriori.

Once steady states are found, HarmonicBalance.jl automatically classifies each state by whether it is real (complex roots bear no physical meaning) and by its stability. The final output is a `Result` object.

3.3 Visualisation

Functionality for static plotting of steady-state and time-dependent solutions is provided using the `Matplotlib` library [88]. At the core of our plotting routines is the function `transform_solutions`, which evaluates a symbolic expression by substituting each of the solutions from a `Result` object. This enables 1D and 2D plots of each solution and functions derived from them. Functionality for selection of transformed solutions (e.g. those with the maximal amplitudes) is available through the method `map_multi_solutions`[7]. We implemented classification functionality to help navigate potentially involved solution landscapes to show/hide/label solutions satisfying a given condition.

---

[7]In general, this function provides support for many-to-one functions of multiple transformed solutions.

Rather than the solutions themselves, a *phase diagram* is often desired, which shows how the qualitative behaviour of the system changes in parameter space. We provide the functionality to distinguish parameter space regions by the number of (stable/all) solutions. We include an additional test that is able to detect qualitative differences between points in parameter space where the number of stable/unstable solutions is invariant (e.g. transcritical bifurcations)[8].

Finally, we highlight the function `interactive_phase_diagram_2D`; this produces an interactive window where the solution landscape can be explored along a given dimension by simply clicking on different parts of a two-dimensional diagram.

3.4 Time-dependent simulations

The behaviour of a system can be found using a numerical ODE solver and an initial condition $\mathbf{u}(T = T_0)$. This approach is beneficial for analysing systems whose parameters are being adiabatically varied in time, i.e. slower than characteristic system reaction times - a common experimental approach to explore the solution landscape, detecting bifurcations and hysteresis. Other potential uses of time-dependent simulations are verifying steady-state results, their stability and fluctuation spectra, and identifying their basins of attraction.

A time-dependent simulation may use either a `DifferentialEquation` or a `HarmonicEquation`. The former represents the system in the time domain and uses no approximations. However, the numerical propagation of oscillatory dynamics can be extremely inefficient. Specifically, time grids need to resolve fractions of the period of the fastest oscillation to keep numerical precision. A `HarmonicEquation` is significantly faster to solve since it describes the system using the slow time T, where the oscillatory steady states appear time-independent. This approach, however, can only capture the chosen set of harmonics and nearby frequencies through the slowly varying amplitudes $\mathbf{u}$. To include additional frequencies, the harmonic ansatz must be expanded.

4 Comparison with other harmonic balance implementations

Steady-state problems in nonlinear periodic ODEs appear in diverse areas of science and technology. Several free and commercial packages exist to solve them. Examples of open-source software include Xyce - a high-performance parallel electronic simulator that can perform harmonic balance analysis [89]. The harmonic balance method is also natively supported for general nonlinear multiphysics finite element simulations in the open-source C++ FEM library Sparselizard [90]. An example of a commercial software package is Cadence AWR Microwave Office, which focuses on the analysis of electrical circuits in the frequency domain (i.e. calculation of voltage and current for a given set of elements), and includes features that are not yet implemented in HarmonicBalance.jl (e.g., noise analysis). Generally, these packages are use-case specialised, making their application in other settings challenging.

Likely the closest existing tool to our Julia package is NLvib [72]. However, it focuses on predefined problems primarily relevant to mechanical engineering and requires a MATLAB license to use, which is unaffordable for many research groups and educational institutes. Crucially, in all of the above packages, the harmonic balance equations are solved by either *i)* time evolution from a given set of initial conditions or *ii)* single-root finding methods such as Newton's. These

---

[8]In short, this function first labels each point in parameter space with a bit string indicating the stability of each solution there (e.g. `[is_stable(sol_1),is_stable(sol_2),···]=[011···]`). Next, each bit string is assigned to a unique integer identifier.

approaches only return one steady state at a time, and even when combined with the continuation of a known solution, disconnected solution branches remain hidden. Our distinction from existing work lies in three main points:

- The use of homotopy continuation allows us to find all possible solutions of the harmonic balance equations, where solutions for multiple parameters can be obtained reliably and efficiently.

- Arbitrary equations of motion can be entered and processed. This is crucial for use by the academic community, where an ab-initio approach to physical problems is often preferred over specialised GUI-based tools.

- The code and its dependencies are entirely open-source. This enables a natural synergy with Julia's existing rich ecosystem for scientific computing.

5 Examples of HarmonicBalance.jl usage

In this Section, we present several example problems. Instructions to install HarmonicBalance.jl and further detailed examples can be found at https://github.com/NonlinearOscillations/HarmonicBalance.jl and in the links therein.

5.1 Duffing oscillator

The simplest use case for HarmonicBalance.jl is a driven Duffing resonator governed by Eq. (5). For a driving frequency ω_d in the vicinity of the natural resonance frequency ω_0, the dominant behaviour can be captured by a single harmonic, the excitation frequency ω_d. The basic lines of code follow below: definition of the system, implementation of the harmonic ansatz

$$x(t) = u_1(T)\cos(\omega_d t) + v_2(T)\sin(\omega_d t), \tag{12}$$

and derivation of corresponding harmonic equations proceed as

```
using HarmonicBalance
@variables α, ωd, ω0, F, t, γ, x(t) # declare constant variables and
    a function x(t)
diff_eq = DifferentialEquation(d(d(x, t),t) + ω0^2 * x + γ*d(x,t)  +
    α*x^3 ~ F*cos(ωd*t), x)
add_harmonic!(diff_eq, x, ωd) # specify the ansatz x = u(T) cos(ωdt)
    + v(T) sin(ωdt)
# implement ansatz to get harmonic equations
harmonic_eq = get_harmonic_equations(diff_eq)
```

harmonic_eq is a HarmonicEquation object, corresponding to Eq. (A.4) in Appendix A. The output of this code is

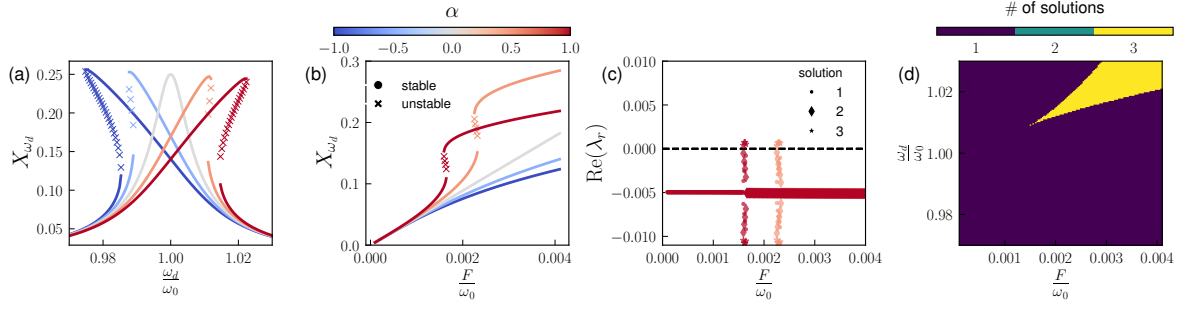
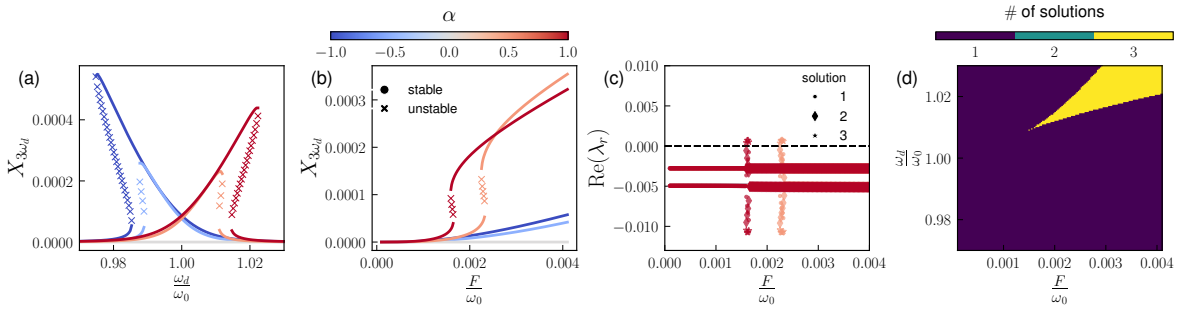
```
A set of 2 harmonic equations
Variables: u1(T), v1(T)
Parameters: α, ω, γ, ω0, F

Harmonic ansatz:
x(t) = u1*cos(ωd*t) + v1*sin(ωd*t)

Harmonic equations:
```

I. Characteristics of the dominant harmonic ($\omega_d \to \omega_d \approx \omega_0$) in the steady state.

II. Characteristics of the superharmonic response ($\omega_d \to 3\omega_d \approx \omega_0$) in the steady state.

Figure 4: **Steady-state solutions for a single-frequency-driven Duffing resonator.** (top) Harmonic amplitudes as a function of (a) the driving frequency ratio ω_d/ω_0 ($F/\omega_0 = 0.0025$) and (b) driving amplitude ratio F/ω_0 (positive detuning, $\omega_d = 1.01\omega_0$). Multiple values of $\alpha \in (-1, 1)$ are color-encoded. (c) Real parts of complex eigenvalues of Jacobian matrix for solutions in (b). Unstable solutions show positive values for at least one of the eigenvalues. (d) Two-dimensional phase diagram depicting the number of steady-state solutions in the $\{F/\omega_0, \omega_d/\omega_0\}$ plane for $\alpha = 1$. Panels at the top (bottom) row correspond to calculation including a single (two) relevant harmonic(s) and plot X_{ω_d} ($X_{3\omega_d}$). For all panels, $\gamma/\omega_0 = 0.01$.

```
(ω0^2)*u1(T) + γ*Differential(T)(u1(T)) + (3//4)*α*(u1(T)^3) + (2//1)*ω*Differential
    (T)(v1(T)) + γ*ω*v1(T) + (3//4)*α*(v1(T)^2)*u1(T) - F - (ω^2)*u1(T) ~ 0

γ*Differential(T)(v1(T)) + (ω0^2)*v1(T) + (3//4)*α*(v1(T)^3) + (3//4)*α*(u1(T)^2)*v1
    (T) - (2//1)*ω*Differential(T)(u1(T)) - (ω^2)*v1(T) - γ*ω*u1(T) ~ 0
```

To calculate the solutions for multiple values of ω_d, we introduce the tuples

```
fixed = (α => 1., ω0 => 1.0, F => 0.0025, γ=> 0.01) # fixed
    parameters
swept = ωd => LinRange(0.97, 1.03, 100) # range of parameter values
```

and simply call

```
solutions = get_steady_states(harmonic_eq, swept, fixed)
```

The steady-state amplitude corresponding to the harmonic ω_d is given by $X_{\omega_d} = \sqrt{u_1^2 + v_1}$; this can be inspected via

```
plot_1D_solutions(solutions, x="ωd", y="sqrt(u1^2 + v1^2)")
```

which produces curves similar to Figs. 4a,b (top row). In such curves, markers are employed to denote the stability of solutions, determined from the Jacobian matrix eigenvalues, which are

evaluated at each point [Fig. 4c]. Real and imaginary parts of Jacobian eigenvalues are displayed via a call to

```
plot_1D_jacobian_eigenvalues(soln_1d, x="ωd");
```

Moreover, simultaneous parametric sweeps over many parameter dimensions can be produced by adding ranges to swept, e.g., to get a 2D dataset,

```
swept = (F =>LinRange(0.0001,0.0041,150), ωd=>LinRange(0.97,
    1.03,150))
```

A 2D phase diagram depicting the total number of solutions, irrespective of stability [Fig. 4d], can then be produced by

```
plot_2D_phase_diagram(solutions,stable=false,observable="nsols")
```

The cubic nonlinearity leads to multiple frequency upconversion processes (Section 2.1). In particular, we focus on the conversion of a resonantly excited oscillation at $\omega_d \approx \omega_0$ to an off-resonant oscillation at $3\omega_d$[9]. The analysis of this mechanism is helpful to study convergence studies of the ansatz. To corroborate it, we implement the ansatz

$$x(t) = u_1(T)\cos(\omega_d t) + v_1(T)\sin(\omega_d t) + u_2(T)\cos(3\omega_d t) + v_2(T)\sin(3\omega_d t), \qquad (13)$$

including harmonics at ω_d and $3\omega_d$:

```
add_harmonic!(diff_eq, x, ωd)
add_harmonic!(diff_eq, x, 3*ωd)
harmonic_eq = get_harmonic_equations(diff_eq)
```

The resulting HarmonicEquation object is printed out

```
A set of 4 harmonic equations
Variables: u1(T), v1(T), u2(T), v2(T)
Parameters: ω, ω0, α, γ, F

Harmonic ansatz:
x(t) = u1*cos(ωt) + v1*sin(ωt) + u2*cos(3ωt) + v2*sin(3ωt)

Harmonic equations:

γ*Differential(T)(u1(T)) + (ω0^2)*u1(T) + (3//4)*α*(u1(T)^3) + γ*ω*v1(T) + (2//1)*ω*
    Differential(T)(v1(T)) + (3//4)*α*(v1(T)^2)*u1(T) + (3//2)*α*(u2(T)^2)*u1(T) +
    (3//2)*α*(v2(T)^2)*u1(T) + (3//4)*α*(u1(T)^2)*u2(T) + (3//2)*α*u1(T)*v1(T)*v2(T)
    - F - (ω^2)*u1(T) - (3//4)*α*(v1(T)^2)*u2(T) ~ 0

γ*Differential(T)(v1(T)) + (ω0^2)*v1(T) + (3//4)*α*(v1(T)^3) + (3//4)*α*(u1(T)^2)*v1
    (T) + (3//4)*α*(u1(T)^2)*v2(T) + (3//2)*α*(u2(T)^2)*v1(T) + (3//2)*α*(v2(T)^2)*
    v1(T) - (2//1)*ω*Differential(T)(u1(T)) - (ω^2)*v1(T) - γ*ω*u1(T) - (3//4)*α*(v1
    (T)^2)*v2(T) - (3//2)*α*u1(T)*u2(T)*v1(T) ~ 0

γ*Differential(T)(u2(T)) + (ω0^2)*u2(T) + (1//4)*α*(u1(T)^3) + (3//4)*α*(u2(T)^3) +
    (6//1)*ω*Differential(T)(v2(T)) + (3//4)*α*(v2(T)^2)*u2(T) + (3//1)*γ*ω*v2(T) +
    (3//2)*α*(u1(T)^2)*u2(T) + (3//2)*α*(v1(T)^2)*u2(T) - (9//1)*(ω^2)*u2(T) -
    (3//4)*α*(v1(T)^2)*u1(T) ~ 0

(ω0^2)*v2(T) + γ*Differential(T)(v2(T)) + (3//4)*α*(v2(T)^3) + (3//4)*α*(u1(T)^2)*v1
    (T) + (3//2)*α*(u1(T)^2)*v2(T) + (3//4)*α*(u2(T)^2)*v2(T) + (3//2)*α*(v1(T)^2)*
    v2(T) - (1//4)*α*(v1(T)^3) - (6//1)*ω*Differential(T)(u2(T)) - (9//1)*(ω^2)*v2(T
    ) - (3//1)*γ*ω*u2(T) ~ 0
```

Steady states can now be retrieved again from

---

[9]The Duffing nonlinearity can also enact the conversion of a response excited below resonance at $\omega_d \approx \omega_0/3$, to a resonant oscillation at $3\omega_d \approx \omega_0$.

```
solutions = get_steady_states(harmonic_eq, swept, fixed)
```

In Fig. 4 bottom, the nonlinearity, yields a finite amplitude $X_{3\omega_d} = \sqrt{u_2^2 + v_2^2} > 0$ for the third harmonic $3\omega_d = \omega_0$. Comparison of phase diagrams for the two ansatzes in Fig. 4d does not reveal new regions in terms of the number of solutions. As described in Appendix A, this insight justifies a perturbative treatment of the upconverted response.

5.2 Navigating involved solution landscapes: Coupled Duffing resonators

The power of the homotopy continuation method lies in its ability to capture all possible solutions as parameters are varied. In contrast, experiments in nonlinear systems often implement parameter variations by adiabatic parameter sweeps, which can only access a subset of steady states. This is because such protocol reaches only one steady state at a time.

The knowledge of all possible solutions can be beneficial to interpret experiments of coupled nonlinear oscillators. As the number of oscillators increases, so does the dimensionality of the problem and the number of potential steady states (Section 2.2) with corresponding intricate solution topology. As an example of how increasingly complex problems are suitable for treatment by HarmonicBalance.jl, we consider in this section the harmonic equations for two coupled Duffing resonators (natural frequencies ω_x and ω_y, coupling k), governed by

$$\ddot{x}(t) + \omega_x^2 x(t) + \gamma \dot{x}(t) + \alpha x(t)^3 - ky(t) = F\cos(\omega_d t), \tag{14}$$

$$\ddot{y}(t) + \omega_y^2 y(t) + \gamma \dot{y}(t) + \alpha y(t)^3 - kx(t) = F\cos(\omega_d t). \tag{15}$$

In addition to obtaining steady states for multiple parameters using homotopy continuation, we solve the time dynamics to simulate an experimentally standard parametric sweep via an interface with DifferentialEquations.jl. This can be attained via the following lines of code

```
@variables ω_x, ω_y, t, ω, F, γ, α, k, x(t), y(t);

free_eq = [ d(d(x, t),t) + ω_x^2 * x +γ*d(x,t) + α*x^3 - k*y,
  d(d(y,t),t) + ω_y^2*y + γ*d(y,t) + α*y^3 - k*x]

forces = [F*cos(ω*t), F*cos(ω*t)]

diff_eq = DifferentialEquation(free_eq - forces, [x, y])

add_harmonic!(diff_eq, x, ω) # x will oscillate at ω
add_harmonic!(diff_eq, y, ω) # y will oscillate at ω
harmonic_eq = get_harmonic_equations(diff_eq)
```

Execution of the above block also prints a detailed summary of the simplified system

```
A set of 4 harmonic equations
Variables: u1(T), v1(T), u2(T), v2(T)
Parameters: α, ω, ω_x, γ, k, F, ω_y

Harmonic ansatz:
x(t) = u1*cos(ωt) + v1*sin(ωt)
y(t) = u2*cos(ωt) + v2*sin(ωt)

Harmonic equations:

(ω_x^2)*u1(T) + γ*Differential(T)(u1(T)) + (3//4)*α*(u1(T)^3) + (2//1)*ω*
    Differential(T)(v1(T)) + γ*ω*v1(T) + (3//4)*α*(v1(T)^2)*u1(T) - F - k*u2(T) - (ω
    ^2)*u1(T) ~ 0
```

```
γ*Differential(T)(v1(T)) + (ω_x^2)*v1(T) + (3//4)*α*(v1(T)^3) + (3//4)*α*(u1(T)^2)*
    v1(T) - (ω^2)*v1(T) - k*v2(T) - (2//1)*ω*Differential(T)(u1(T)) - γ*ω*u1(T) ~ 0

γ*Differential(T)(u2(T)) + (ω_y^2)*u2(T) + (3//4)*α*(u2(T)^3) + (2//1)*ω*
    Differential(T)(v2(T)) + γ*ω*v2(T) + (3//4)*α*(v2(T)^2)*u2(T) - F - k*u1(T) - (ω
    ^2)*u2(T) ~ 0

(ω_y^2)*v2(T) + γ*Differential(T)(v2(T)) + (3//4)*α*(v2(T)^3) + (3//4)*α*(u2(T)^2)*
    v2(T) - k*v1(T) - (ω^2)*v2(T) - (2//1)*ω*Differential(T)(u2(T)) - γ*ω*u2(T) ~ 0
```

Multi-parameter steady states can be obtained,

```
fixed_parameters = (ω_x=> 1., ω_y=> 1.05, γ=> 2E-3, F=> 1E-2, α=> 1E
    -1, k=> 5E-2])
sweep = (ω => LinRange(0.85,1.25,150))
solutions = get_steady_states(problem, sweep, fixed)
```

This yields a `Result` object,

```
Solution branches:   11
    of which real:    7
    of which stable:  4
```

We visualise the amplitudes of stable steady-state solutions via

```
plot_1D_solutions(soln, x="ω", y="sqrt(u1^2 + v1^2)", plot_only=["
    physical", "stable"],filename="coupled_duffing_sols");
```

Here we saved results to a `.jld2` file by filling the `filename` keyword argument. Data can be subsequently recovered via use of the `load` function.

We are now in a position to check the accessibility of the steady states by performing a backward, time-dependent sweep of the frequency, initiating at a given particular condition:

```
# select a solution and evolve from it
    s1 = get_single_solution(soln, branch=5, index=75);

    sweep = HarmonicBalance.TimeEvolution.ParameterSweep(ω =>
        (1.2, 0.9), (0, 1E5)) #HarmonicBalance.TimeEvolution.
        ParameterSweep(ω => (s1[ω], 0.95), (0, 1E5))

    function t_solve(s) """retrieve long-time evolved results
        from the slow-flow equation"""
        problem = HarmonicBalance.TimeEvolution.ODEProblem(
            averagedEOM, steady_solution=s, timespan=(0,1E5),
            sweep=sweep)
        time_soln = HarmonicBalance.TimeEvolution.solve(problem,
            saveat=10);
    end
    time_soln_1 = t_solve(s1);
```

A forward sweep can also be performed by setting

```
sweep = HarmonicBalance.TimeEvolution.ParameterSweep(ω =>
    (0.9,1.2), (0, 1E5))
time_soln_2 = t_solve(s1);
```

Large-time amplitudes from backwards/forward time sweeps can be retrieved from

```
X1 = sqrt.(getindex.(time_soln_1.u,1).^2 .+ getindex.(time_soln_1.u
    ,2).^2)
X2 = sqrt.(getindex.(time_soln_2.u,1).^2 .+ getindex.(time_soln_2.u
    ,2).^2)
```

Results from these sweeps are presented along with the steady states in Fig. 5.

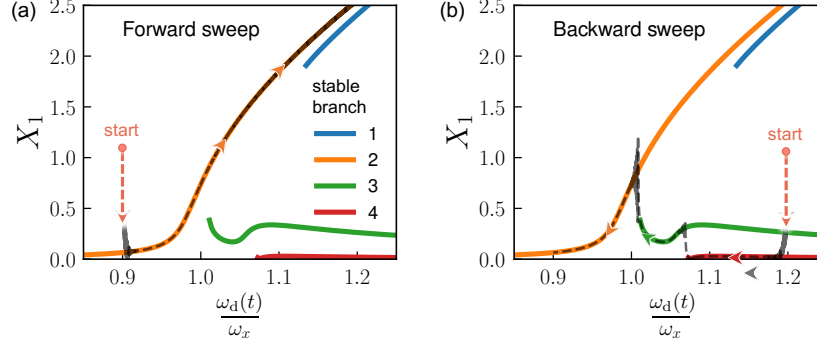



Figure 5: **Time-dependent driving frequency sweep (gray) and stable steady-state landscape (colored) in two linearly coupled, driven Duffing resonators.** Steady-state solution amplitudes $X_1 = \sqrt{u_1^2 + v_1^2}$ for a system initiated away from any steady state, rapidly collapsing into one of the available solutions. Adiabatic variation of the driving frequency follows a subset of stable, steady states. The arrows indicate the direction of the sweep; jumps occur at points where a solution branch ceases to be stable. The observed behaviour is hysteretic, i.e., it depends on the sweep direction. The parameters chosen are $\omega_x = 1, \omega_y = 1.05, \gamma = 2 \cdot 10^{-3}, F = 10^{-2}, \alpha = 10^{-1}, J = 5 \cdot 10^{-2}$. The sweeping time is $\tau = 10^5 \gg \gamma^{-1}$, much slower than any relaxation time to minimise non-adiabatic effects.

5.3 Increasing computational complexity: Performance scaling

Here, we shortly illustrate the performance of HarmonicBalance.jl for varying system sizes. We consider a chain of linearly coupled Duffing oscillators, each with nonlinear damping (amplitude η, other parameters defined above Eq. (6)),

$$\ddot{x}_i(t) + \omega_0^2 x_i(t) + \alpha x_i(t)^3 + \eta x_i(t)^2 \dot{x}_i(t) - k \sum_{j=i\pm 1} x_j(t) = F\cos(\omega t), \quad i = 1, 2, ..., N. \quad (16)$$

The displacement of each oscillator, x_i, is expanded using the harmonic oscillating at ω, giving the harmonic ansatz

$$x_i(t) = u_i(T)\cos(\omega t) + v_i(T)\sin(\omega t). \quad (17)$$

As explained in Section 2, each oscillator adds 2 harmonic equations of order 3. Therefore, a chain of length N and $M = 1$ leads to a set of $2N$ equations. The corresponding Bézout bound on the number of solutions, and hence the number of paths which must be tracked by the homotopy continuation algorithm, is 3^{2N}.

In Table. 2, we show the computational times necessary for symbolic manipulation and homotopy solving as well as the number of complex and real solutions found. We highlight solving a chain of $N = 5$ resonators still takes a few minutes on a single CPU. Notably, the number of unique solutions is significantly smaller than the Bézout bound in all cases, with the number of real solutions being smaller still. The bottleneck is, therefore, the initial tracking of the homotopy from 3^{2N} paths to the relatively few non-singular ones. Subsequent steps such as solving for multiple parameter sets (i.e. tracking parameter homotopy paths) and determining stability (Jacobian evaluation and diagonalisation) only involve non-singular paths and thus are relatively inexpensive.

Two remarks regarding performance are in order. First, the process of path tracking is naturally well-suited for parallelisation, and HomotopyContinuation.jl does include the necessary functionality. Second, systems such as the Duffing chain possess spatial symmetries (in this case, inversion

N	1	2	3	4	5
t_{symbolic} [s]	0.16	0.38	0.71	1.11	1.54
t_{solve} [s]	0.52	1.47	17.3	173	1801
Bézout bound	9	81	729	6561	59049
complex solutions	3	11	59	545	3577
real solutions	3	11	32	103	310

Table 2: **Finding the steady states of Eq.** (16) **for varying chain length** N**.** The rows describe (i) Computational time required to obtain the harmonic equations and the symbolic Jacobian t_{symbolic}. (ii) Time to solve the harmonic equations for 50 parameter sets t_{solve}. (iii) The Bézout bound. (iv) Number of complex solutions found. (v) Number of real solutions found. The parameters used are $\omega_0 = 1, \alpha = 1, k = 0.1, \eta = 0.1, F = 0.01$. A single core of an Intel i7-8550U CPU was used.

symmetry) and internal/gauge symmetries (e.g. discrete time translation, $u_i \mapsto -v_i, v_i \mapsto u_i$) which cause some solutions to be degenerate. Making use of this property can further reduce the computational time needed.

6   Conclusion

In this work, we have introduced a Julia package to treat the steady-state behaviour of generic nonlinear, polynomial, dynamical systems with harmonic time dependence. Combining symbolic and numerical calculations with simple graphical capabilities, our package is developed with simplicity of use in mind while enabling the study of complex nonlinear systems.

Its modular design paves the way for future methodological extensions, including detection of Hopf bifurcations [91], limit cycles and chaos [92], higher-order Krylov-Bogoliubov averaging method [93], as well as interfaces with existing dedicated libraries to treat nonlinear spatially-extended or quantum systems [94, 95]. Its usage can assist a breadth of fields, where nonlinear harmonically-driven systems appear, such as modal analysis in structural dynamics [6, 28], electric circuits [7, 29, 30], nonlinear optics [8, 31–37], optomechanics [17, 38–40], micro- and nanome-chanics [41–50], oscillator networks [51–56], Ising machines [57–62], and many-body light-matter systems [63–69].

We hope that the free availability of a user-friendly, high-performance code for harmonic nonlinear dynamics calculations will help advance multiple disciplines by allowing researchers to perform computations currently considered out of reach due to their complexity. In addition, we hope its implementation, sitting on Julia's rich ecosystem for high-performance scientific computing, will help nurture a multidisciplinary open-source community.

Acknowledgements

Author contributions   J.K. and J.d.P. developed the package, T.H. provided insight on the theoretical approach, and O.Z. supervised the project. All authors contributed to the interpretation of results and writing of the manuscript.

Funding information J.K., T.H. were supported by the Swiss National Science Foundation through grant CRSII5 177198/1. J.d.P. was supported by the ETH Fellowship program (grant no. 20-2 FEL-66). O.Z. acknowledges funding through SNSF grant PP00P2_190078 and from the Deutsche Forschungsgemeinschaft (DFG) - project number 449653034.

A Harmonic equations for a Duffing resonator

Here, we derive the harmonic equations for a single Duffing resonator, governed by the equation

$$\ddot{x}(t) + \omega_0^2 x(t) + \alpha x^3(t) = F\cos(\omega_d t + \theta), \tag{A.1}$$

where we considered driving with a finite phase offset θ for generality. As explained in Section 2, for a periodic driving at frequency ω_d and a weak nonlinearity α, we expect the response at frequency ω_d to be dominant, followed by a response at $3\omega_d$ due to frequency conversion.

Expanding in ω_d only

We first describe the established [3,70,75] approach to finding the steady states of Eq. (A.1), where frequency conversion is only added perturbatively. The starting point is a harmonic ansatz for x of the form Eq. (7), containing a single frequency ω_d,

$$x(t) = u(T)\cos(\omega_d t) + v(T)\sin(\omega_d t), \tag{A.2}$$

with the harmonic variables u and v. The *slow time T* is, for now, equivalent to t. Substituting this ansatz into Eq. (A.1) results in

$$
\begin{aligned}
&\left[\ddot{u} + 2\omega_d\dot{v} + u\left(\omega_0^2 - \omega_d^2\right) + \frac{3\alpha\left(u^3 + uv^2\right)}{4} + F\cos\theta\right]\cos(\omega_d t) \\
&+ \left[\ddot{v} - 2\omega_d\dot{u} + v\left(\omega_0^2 - \omega_d^2\right) + \frac{3\alpha\left(v^3 + u^2 v\right)}{4} - F\sin\theta\right]\sin(\omega_d t) \\
&+ \frac{\alpha\left(u^3 - 3uv^2\right)}{4}\cos(3\omega_d t) + \frac{\alpha\left(3u^2 v - v^3\right)}{4}\sin(3\omega_d t) = 0.
\end{aligned}
\tag{A.3}
$$

We see that x^3 in Eq. (A.1) generates terms that oscillate at $3\omega_d$, describing the process of frequency upconversion. We now Fourier-transform both sides of Eq. (A.3) with respect to ω_d to obtain the harmonic equations. This process is equivalent to extracting the respective coefficients of $\cos(\omega_d t)$ and $\sin(\omega_d t)$. Here the distinction between t and T becomes important: since the evolution of $u(T)$ and $v(T)$ is assumed to be slow, they are treated as constant for the purpose of Fourier transformation. Since we are interested in steady states, we drop the higher-order derivatives and rearrange the resulting equation to the form of Eq. (8) of the main text

$$\frac{d}{dT}\begin{pmatrix} u \\ v \end{pmatrix} = \frac{1}{8\omega_d}\begin{pmatrix} 4v\left(\omega_0^2 - \omega_d^2\right) + 3\alpha\left(v^3 + u^2 v\right) - 4F\sin\theta \\ 4u\left(\omega_d^2 - \omega_0^2\right) - 3\alpha\left(u^3 + uv^2\right) - 4F\cos\theta \end{pmatrix}. \tag{A.4}$$

Note that our assumption that $u(T)$ and $v(T)$ are slowly changing, i.e. composed of small frequency terms also sets constraints in Fourier space: we neglect all the frequencies which are not close to ω_d. In the extreme case of constant u and v, the described frequency range reduces to the discrete

frequency ω_d. Steady states can now be found by setting the l.h.s. to zero, i.e., assuming $u(T)$ and $v(T)$ constant and neglecting any transient behaviour. This step is referred to in the literature as "balancing the harmonics" [72]. This results in a set of 2 nonlinear polynomial equations of order 3, for which the maximum number of solutions set by Bézout theorem is $3^2 = 9$. Depending on the parameters, the number of real solutions is known to be between 1 and 3, see the solution diagrams Section 5.

The steady states describe a response that may be recast as $x_0(t) = X_0 \cos(\omega_d t + \phi)$, where $X_0 = \sqrt{u^2 + v^2}$ and $\phi = -\text{atan}(v/u)$. Frequency conversion from ω_d to $3\omega_d$ can be found by setting $x(t) \equiv x_0(t) + \delta x(t)$ with $|\delta x(t)| \ll |x_0(t)|$ and expanding Eq. (A.1) to first-order in $\delta x(t)$. The resulting equation

$$\delta \ddot{x}(t) + \left[\omega_0^2 + \frac{3\alpha X_0^2}{4} \right] \delta x(t) = -\frac{\alpha X_0^3}{4} \cos(3\omega_d t + 3\phi), \tag{A.5}$$

describes a simple harmonic oscillator, which is exactly soluble. Correspondingly, a response of $\delta x(t)$ at frequency $3\omega_d$ is observed. Since this response is obtained perturbatively for each steady state of Eq. (A.4), no previously-unknown solutions are generated in the process.

Expanding in ω_d and $3\omega_d$

An approach in the spirit of harmonic balance is to use both harmonics ω_d and $3\omega_d$ on the same footing, i.e., to insert the ansatz

$$x(t) = u_1(T) \cos(\omega_d t) + v_1(T) \sin(\omega_d t) + u_2(T) \cos(3\omega_d t) + v_2(T) \sin(3\omega_d t), \tag{A.6}$$

with u_1, u_2, v_1, v_2 being the harmonic variables. As before we substitute the ansatz into Eq. (A.1), drop second derivatives with respect to T and Fourier-transform both sides. Now, the respective coefficients correspond to $\cos(\omega_d t)$, $\sin(\omega_d t)$, $\cos(3\omega_d t)$ and $\sin(3\omega_d t)$. Rearranging, we obtain

$$\frac{du_1}{dT} = \frac{1}{2\omega_d} \left[\left(\omega_0^2 - \omega_d^2 \right) v_1 + \frac{3\alpha}{4} \left(v_1^3 + u_1^2 v_1 + u_1^2 v_2 - v_1^2 v_2 + 2u_2^2 v_1 + 2v_2^2 v_1 - 2u_1 u_2 v_1 \right) + F \sin\theta \right],$$

$$\frac{dv_1}{dT} = \frac{1}{2\omega_d} \left[\left(\omega_d^2 - \omega_0^2 \right) u_1 - \frac{3\alpha}{4} \left(u_1^3 + u_1^2 u_2 + v_1^2 u_1 - v_1^2 u_2 + 2u_2^2 u_1 + 2v_2^2 u_1 + 2u_1 v_1 v_2 \right) - F \cos\theta \right],$$

$$\frac{du_2}{dT} = \frac{1}{6\omega_d} \left[\left(\omega_0^2 - 9\omega_d^2 \right) v_2 + \frac{\alpha}{4} \left(-v_1^3 + 3v_2^3 + 3u_1^2 v_1 + 6u_1^2 v_2 + 3u_2^2 v_2 + 6v_1^2 v_2 \right) \right],$$

$$\frac{dv_2}{dT} = \frac{1}{6\omega_d} \left[\left(9\omega_d^2 - \omega_0^2 \right) u_2 - \frac{\alpha}{4} \left(u_1^3 + 3u_2^3 + 6u_1^2 u_2 - 3v_1^2 u_1 + 3v_2^2 u_2 + 6v_1^2 u_2 \right) \right]. \tag{A.7}$$

In contrast to Eqs. (A.4), we now have 4 equations of order 3, allowing up to $3^4 = 81$ solutions (the number of unique real ones is again generally far smaller, as seen in Section 5). The larger number of solutions is explained by higher harmonics which cannot be captured perturbatively by the single-frequency ansatz. In particular, those where the $3\omega_d$ component is significant. Such solutions appear, e.g., for $\omega_d \approx \omega_0/3$ where the generated $3\omega_d$ harmonic is close to the natural resonance frequency.

B   Solving steady states for multiple parameter values

Evaluating Eq. (8) at each parameter set determines a distinct numerical system of polynomial equations, whose roots can be found using homotopy continuation; this generally requires tracking the maximum number of roots given by Bézout's theorem, which may be very time-consuming. Having solved for one parameter set, the entire tracking procedure is not necessary for others - one only needs to track paths leading to non-singular solutions. In particular, we carry out

1. A "warm-up", i.e. path tracking from the roots of, e.g., $\mathbf{U}(\mathbf{u}_0) = \{u_1^{d_1}-1, v_1^{d_1}-1, u_2^{d_2}-1, v_2^{d_2}-1, \cdots\}$ $(d_r, r = (1, 2, \cdots, NM)$ denotes the degree of the polynomial $\bar{F}_r(\mathbf{u}_0))$ towards the roots of a "generic" system, namely $\bar{\mathbf{F}}(\mathbf{u}_0)$ evaluated with random complex (unphysical) parameters[10]. All singular paths are filtered out during this procedure.

2. The number of complex roots of the generic system is an upper bound for the number of roots of $\bar{\mathbf{F}}(\mathbf{u}_0) = 0$ for arbitrary parameters [25]. In a final step, we apply a *parameter homotopy* (a mapping between the collection of systems $\bar{\mathbf{F}}(\mathbf{u}_0) = 0$ obtained by only varying the parameters) towards a physical real parameter value. This may involve tracking a massively reduced number of paths, hence being much faster than step 1.

3. Repeat step 2 for each parameter value.

Despite the computational overhead of the problem's "complexification", complex solution paths allow for a stable path tracking algorithm, preventing singularities that will otherwise show up in a real-space parameter homotopy (see Ref. [25] for details).

References

[1] J. J. Stoker, *Nonlinear Vibrations in Mechanical and Electrical Systems*, Interscience (1950).

[2] S. Strogatz, *Nonlinear Dynamics and Chaos: With Applications to Physics, Biology, Chemistry, and Engineering*, CRC Press, doi:https://doi.org/10.1201/9780429492563 (2015).

[3] R. H. Rand, *Lecture Notes on Nonlinear Vibrations*, http://audiophile.tam.cornell.edu/randdocs/nlvibe52.pdf (2005).

[4] G. Falkovich, *Fluid mechanics: A short course for physicists*, Cambridge University Press, doi:10.1017/CBO9780511794353 (2011).

[5] S. B. Niku, *Introduction to robotics: analysis, control, applications*, John Wiley & Sons (2020).

[6] D. J. Ewins, *Modal Testing: Theory, Practice and Application (Mechanical Engineering Research Studies: Engineering Dynamics Series)*, Wiley (2000).

[7] U. L. Rohde, A. K. Poddar and G. Böck, *The Design of Modern Microwave Oscillators for Wireless Applications*, John Wiley & Sons, Inc., Hoboken, NJ, USA, doi:10.1002/0471727172 (2005).

---

[10]For some classes of polynomials, alternative starting systems (see polyhedral homotopy, implemented in Homotopy-Continuation.jl [26, 46]) can significantly decrease the number of solution paths to be tracked compared with a total degree homotopy.

[8] Y. R. Shen, *The Principles of Nonlinear Optics (Wiley Classics Library)*, New York, Wiley-Interscience, 1984, 575 p. (2002).

[9] J. Hofbauer, K. Sigmund *et al.*, *Evolutionary games and population dynamics*, Cambridge University Press, doi:10.1017/CBO9781139173179 (1998).

[10] I. R. Epstein and J. A. Pojman, *An Introduction to Nonlinear Chemical Dynamics: Oscillations, Waves, Patterns, and Chaos* , Oxford University Press, doi:10.1093/oso/9780195096705.001.0001 (1998).

[11] M. Volkenstein, *Molecular biophysics*, Elsevier (2012).

[12] J. A. Richards, *Analysis of periodically time-varying systems*, Springer, doi:10.1007/978-3-642-81873-8 (1983).

[13] A. Leuch, L. Papariello, O. Zilberberg, C. L. Degen, R. Chitra and A. Eichler, *Parametric Symmetry Breaking in a Nonlinear Resonator*, Phys. Rev. Lett. **117**, 214101 (2016), doi:10.1103/PhysRevLett.117.214101.

[14] T. L. Heugel, M. Biondi, O. Zilberberg and R. Chitra, *Quantum Transducer Using a Parametric Driven-Dissipative Phase Transition*, Phys. Rev. Lett. **123**, 173601 (2019), doi:10.1103/PhysRevLett.123.173601.

[15] G. Margiani, S. Guerrero, T. L. Heugel, C. Marty, R. Pachlatko, T. Gisler, G. D. Vukasin, H.-K. Kwon, J. M. Miller, N. E. Bousse, T. W. Kenny, O. Zilberberg *et al.*, *Fluctuating Trajectories and Switching Rates of a Synthetic Two-Level System*, arXiv:2112.03357.

[16] R. Chitra and O. Zilberberg, *Dynamical many-body phases of the parametrically driven, dissipative Dicke model*, Phys. Rev. A **92**, 023815 (2015), doi:10.1103/PhysRevA.92.023815.

[17] T. F. Roque, F. Marquardt and O. M. Yevtushenko, *Nonlinear dynamics of weakly dissipative optomechanical systems*, New J. Phys. (2020), doi:10.1088/1367-2630/ab6522.

[18] V. I. Arnold, *Mathematical Methods of Classical Mechanics*, Springer, doi:10.1007/978-1-4757-1693-1 (1978).

[19] C. Cohen-Tannoudji, B. Diu and F. Laloe, *Quantum Mechanics, Volume 2*, Quantum Mechanics (1986).

[20] M. J. Bhaseen, J. Mayoh, B. D. Simons and J. Keeling, *Dynamics of nonequilibrium Dicke models*, Phys. Rev. A **85**, 013817 (2012), doi:10.1103/PhysRevA.85.013817.

[21] S. Aldana, C. Bruder and A. Nunnenkamp, *Equivalence between an optomechanical system and a Kerr medium*, Phys. Rev. A **88**(4) (2013), doi:10.1103/PhysRevA.88.043826.

[22] D. N. Bernshtein, *The number of roots of a system of equations*, Functional Analysis and its applications **9**(3), 183 (1975).

[23] B. Sturmfels, *Solving systems of polynomial equations*, 97. American Mathematical Soc. (2002).

[24] D. A. Cox, J. Little and D. O'Shea, *Using Algebraic Geometry*, pp. 26–76, Springer, ISBN 978-0-387-98487-2, doi:10.1007/978-1-4757-6911-1 (2005).

[25] A. J. Sommese and C. W. Wampler, *The Numerical Solution of Systems of Polynomials Arising in Engineering and Science*, World Scientific, doi:10.1142/5763 (2005).

[26] P. Breiding and S. Timme, *HomotopyContinuation. jl: A package for homotopy continuation in Julia*, In *International Congress on Mathematical Software*, pp. 458–465. Springer, doi:10.1007/978-3-319-96418-8_54 (2018).

[27] S. Timme, *Numerical nonlinear algebra*, Ph.D. thesis, Technische Universität Berlin (Germany) (2021).

[28] G. Kerschen, K. Worden, A. F. Vakakis and J. C. Golinval, *Past, present and future of nonlinear system identification in structural dynamics*, Mech. Syst. Signal Process. (2006), doi:10.1016/j.ymssp.2005.04.008.

[29] E. Rubiola, *Phase Noise and Frequency Stability in Oscillators*, Cambridge University Press, Cambridge, doi:10.1017/CBO9780511812798 (2008).

[30] A. Fallis and S. A. Maas, *Nonlinear microwave and RF circuits*, Artech House (2003).

[31] H. Haken, *Cooperative phenomena in systems far from thermal equilibrium and in nonphysical systems*, Rev. Mod. Phys. (1975), doi:10.1103/RevModPhys.47.67.

[32] P. Del'Haye, A. Schliesser, O. Arcizet, T. Wilken, R. Holzwarth and T. J. Kippenberg, *Optical frequency comb generation from a monolithic microresonator*, Nature **450**(7173), 1214 (2007), doi:10.1038/nature06401.

[33] S. R. Rodriguez, A. Amo, I. Sagnes, L. Le Gratiet, E. Galopin, A. Lemaître and J. Bloch, *Interaction-induced hopping phase in driven-dissipative coupled photonic microcavities*, Nat. Commun. **7**, 11887 (2016), doi:10.1038/ncomms11887.

[34] J. del Pino, F. J. Garcia-Vidal and J. Feist, *Exploiting Vibrational Strong Coupling to Make an Optical Parametric Oscillator Out of a Raman Laser*, Phys. Rev. Lett. **117**(27), 277401 (2016), doi:10.1103/PhysRevLett.117.277401.

[35] D. L. Sounas, J. Soric and A. Alù, *Broadband passive isolators based on coupled nonlinear resonances*, Nat. Electron. **1**, 113 (2018), doi:10.1038/s41928-018-0025-0.

[36] N. Carlon Zambon, S. R. K. Rodriguez, A. Lemaître, A. Harouri, L. Le Gratiet, I. Sagnes, P. St-Jean, S. Ravets, A. Amo and J. Bloch, *Parametric instability in coupled nonlinear microcavities*, Phys. Rev. A **102**, 023526 (2020), doi:10.1103/PhysRevA.102.023526.

[37] K. J. H. Peters and S. R. K. Rodriguez, *Limit Cycles and Chaos Induced by a Nonlinearity with Memory*, arXiv:2108.02680.

[38] M. Aspelmeyer, T. J. Kippenberg and F. Marquardt, *Cavity optomechanics*, Rev. Mod. Phys. **86**, 1391 (2014), doi:10.1103/RevModPhys.86.1391.

[39] K. Pelka, V. Peano and A. Xuereb, *Chimera states in small optomechanical arrays*, Phys. Rev. Research **2**, 013201 (2020), doi:10.1103/PhysRevResearch.2.013201.

[40] R. Burgwal, J. del Pino and E. Verhagen, *Comparing nonlinear optomechanical coupling in membrane-in-the-middle and single-cavity systems*, New Journal of Physics **22**(11), 113006 (2020), doi:10.1088/1367-2630/abc1c8.

[41] M. Poot and H. S. van der Zant, *Mechanical systems in the quantum regime*, Physics Reports **511**(5), 273 (2012), doi:https://doi.org/10.1016/j.physrep.2011.12.004.

[42] L. Papariello, O. Zilberberg, A. Eichler and R. Chitra, *Ultrasensitive hysteretic force sensing with parametric nonlinear oscillators*, Phys. Rev. E **94**(2), 022201 (2016), doi:10.1103/PhysRevE.94.022201.

[43] F. Yang, F. Hellbach, F. Rochau, W. Belzig, E. M. Weig, G. Rastelli and E. Scheer, *Persistent Response in an Ultrastrongly Driven Mechanical Membrane Resonator*, Phys. Rev. Lett. **127**(1), 014304 (2021), doi:10.1103/PhysRevLett.127.014304.

[44] F. Yang, M. Fu, B. Bosnjak, R. H. Blick, Y. Jiang and E. Scheer, *Mechanically Modulated Sideband and Squeezing Effects of Membrane Resonators*, Phys. Rev. Lett. **127**, 184301 (2021), doi:10.1103/PhysRevLett.127.184301.

[45] Z. Mohammadi, T. L. Heugel, J. M. L. Miller, D. D. Shin, H.-K. Kwon, T. W. Kenny, R. Chitra, O. Zilberberg and L. G. Villanueva, *On the effect of linear feedback and parametric pumping on a resonator's frequency stability*, New Journal of Physics **22**(9), 093049 (2020), doi:10.1088/1367-2630/abb1dd.

[46] J. S. Huber, G. Rastelli, M. J. Seitner, J. Kölbl, W. Belzig, M. I. Dykman and E. M. Weig, *Spectral Evidence of Squeezing of a Weakly Damped Driven Nanomechanical Mode*, Phys. Rev. X **10**, 021066 (2020), doi:10.1103/PhysRevX.10.021066.

[47] Z. Mohammadi, T. L. Heugel, J. M. L. Miller, D. D. Shin, H.-K. Kwon, T. W. Kenny, R. Chitra, O. Zilberberg and L. G. Villanueva, *On the effect of linear feedback and parametric pumping on a resonator's frequency stability*, New Journal of Physics **22**(9), 093049 (2020), doi:10.1088/1367-2630/abb1dd.

[48] A. Eichler, T. L. Heugel, A. Leuch, C. L. Degen, R. Chitra and O. Zilberberg, *A parametric symmetry breaking transducer*, Appl. Phys. Lett. **112**(23), 233105 (2018), doi:10.1063/1.5031058.

[49] J. Košata, O. Zilberberg, C. L. Degen, R. Chitra and A. Eichler, *Spin Detection via Parametric Frequency Conversion in a Membrane Resonator*, Phys. Rev. Applied **14**, 014042 (2020), doi:10.1103/PhysRevApplied.14.014042.

[50] D. Hälg, T. Gisler, E. C. Langman, S. Misra, O. Zilberberg, A. Schliesser, C. L. Degen and A. Eichler, *Strong parametric coupling between two ultra-coherent membrane modes*, arXiv:2109.11943.

[51] T. Ozawa, H. M. Price, N. Goldman, O. Zilberberg and I. Carusotto, *Synthetic dimensions in integrated photonics: From optical isolation to four-dimensional quantum Hall physics*, Phys. Rev. A **93**, 043827 (2016), doi:10.1103/PhysRevA.93.043827.

[52] T. L. Heugel, M. Oscity, A. Eichler, O. Zilberberg and R. Chitra, *Classical Many-Body Time Crystals*, Phys. Rev. Lett. **123**, 124301 (2019), doi:10.1103/PhysRevLett.123.124301.

[53] T. Ozawa, H. M. Price, A. Amo, N. Goldman, M. Hafezi, L. Lu, M. C. Rechtsman, D. Schuster, J. Simon, O. Zilberberg and I. Carusotto, *Topological photonics*, Rev. Mod. Phys. **91**(1), 015006 (2019), doi:10.1103/RevModPhys.91.015006.

[54] J. Košata and O. Zilberberg, *Second-order topological modes in two-dimensional continuous media*, Phys. Rev. Research **3**, L032029 (2021), doi:10.1103/PhysRevResearch.3.L032029.

[55] J. del Pino, J. J. Slim and E. Verhagen, *Non-Hermitian chiral phononics through optomechanically-induced squeezing*, arXiv preprint arXiv:2110.14710 (2021).

[56] H. Price, Y. Chong, A. Khanikaev, H. Schomerus, L. J. Maczewsky, M. Kremer, M. Heinrich, A. Szameit, O. Zilberberg, Y. Yang, B. Zhang, A. Alù *et al.*, *Roadmap on Topological Photonics*, arXiv:2201.06315.

[57] Z. Wang, A. Marandi, K. Wen, R. L. Byer and Y. Yamamoto, *Coherent Ising machine based on degenerate optical parametric oscillators*, Phys. Rev. A **88**, 063853 (2013), doi:10.1103/PhysRevA.88.063853.

[58] L. Bello, M. Calvanese Strinati, E. G. Dalla Torre and A. Pe'er, *Persistent Coherent Beating in Coupled Parametric Oscillators*, Phys. Rev. Lett. **123**, 083901 (2019), doi:10.1103/PhysRevLett.123.083901.

[59] M. Calvanese Strinati, L. Bello, A. Pe'er and E. G. Dalla Torre, *Theory of coupled parametric oscillators beyond coupled Ising spins*, Phys. Rev. A **100**, 023835 (2019), doi:10.1103/PhysRevA.100.023835.

[60] M. Frimmer, T. L. Heugel, Ž. Nosan, F. Tebbenjohanns, D. Hälg, A. Akin, C. L. Degen, L. Novotny, R. Chitra, O. Zilberberg and A. Eichler, *Rapid Flipping of Parametric Phase States*, Phys. Rev. Lett. **123**, 254102 (2019), doi:10.1103/PhysRevLett.123.254102.

[61] M. C. Strinati, I. Aharonovich, S. Ben-Ami, E. G. D. Torre, L. Bello and A. Pe'er, *Coherent dynamics in frustrated coupled parametric oscillators*, New Journal of Physics **22**(8), 085005 (2020), doi:10.1088/1367-2630/aba573.

[62] T. L. Heugel, O. Zilberberg, C. Marty, R. Chitra and A. Eichler, *Ising machines with strong bilinear coupling*, arXiv:2103.02625.

[63] A. Griffin, D. W. Snoke and S. Stringari, *Bose-Einstein condensation*, Cambridge University Press, doi:10.1017/CBO9780511524240 (1996).

[64] I. Carusotto and C. Ciuti, *Quantum fluids of light*, Rev. Mod. Phys. **85**, 299 (2013), doi:10.1103/RevModPhys.85.299.

[65] M. Soriente, T. Donner, R. Chitra and O. Zilberberg, *Dissipation-Induced Anomalous Multicritical Phenomena*, Phys. Rev. Lett. **120**, 183603 (2018), doi:10.1103/PhysRevLett.120.183603.

[66] P. Kirton, M. M. Roses, J. Keeling and E. G. Dalla Torre, *Introduction to the Dicke Model: From Equilibrium to Nonequilibrium, and Vice Versa*, Advanced Quantum Technologies **2**(1-2), 1800043 (2019), doi:https://doi.org/10.1002/qute.201800043.

[67] M. Soriente, R. Chitra and O. Zilberberg, *Distinguishing phases using the dynamical response of driven-dissipative light-matter systems*, Phys. Rev. A **101**, 023823 (2020), doi:10.1103/PhysRevA.101.023823.

[68] M. Soriente, T. L. Heugel, K. Arimitsu, R. Chitra and O. Zilberberg, *Distinctive class of dissipation-induced phase transitions and their universal characteristics*, Phys. Rev. Research **3**, 023100 (2021), doi:10.1103/PhysRevResearch.3.023100.

[69] F. Ferri, R. Rosa-Medina, F. Finger, N. Dogra, M. Soriente, O. Zilberberg, T. Donner and T. Esslinger, *Emerging Dissipative Phases in a Superradiant Quantum Gas with Tunable Decay*, Phys. Rev. X **11**, 041046 (2021), doi:10.1103/PhysRevX.11.041046.

[70] R. Lifshitz and M. C. Cross, *Nonlinear Dynamics of Nanomechanical and Micromechanical Resonators*, chap. 1, pp. 1–52, John Wiley & Sons, Ltd, ISBN 9783527626359, doi:https://doi.org/10.1002/9783527626359.ch1 (2008).

[71] J. Guckenheimer and P. Holmes, *Nonlinear oscillations, dynamical systems, and bifurcations of vector fields*, Springer, doi:10.1007/978-1-4612-1140-2 (2013).

[72] M. Krack and J. Gross, *Harmonic balance for nonlinear vibration problems*, Springer, doi:10.1007/978-3-030-14023-6 (2019).

[73] A. C. Luo and J. Huang, *Approximate solutions of periodic motions in nonlinear systems via a generalized harmonic balance*, Journal of Vibration and Control **18**(11), 1661 (2012), doi:10.1177/1077546311421053.

[74] C. W. Gardiner and P. Zoller, *Quantum noise: a handbook of Markovian and non-Markovian quantum stochastic methods with applications to quantum optics*, Springer (2004).

[75] A. H. Nayfeh, *Perturbation methods*, John Wiley & Sons, doi:https://doi.org/10.1002/9783527617609 (2000).

[76] T. Mikami, S. Kitamura, K. Yasuda, N. Tsuji, T. Oka and H. Aoki, *Brillouin-Wigner theory for high-frequency expansion in periodically driven systems: Application to Floquet topological insulators*, Phys. Rev. B **93**, 144307 (2016), doi:10.1103/PhysRevB.93.144307.

[77] A. Eckardt and E. Anisimovas, *High-frequency approximation for periodically driven quantum systems from a Floquet-space perspective*, New Journal of Physics **17**(9), 093039 (2015), doi:10.1088/1367-2630/17/9/093039.

[78] M. S. Rudner and N. H. Lindner, *Band structure engineering and non-equilibrium dynamics in Floquet topological insulators*, Nat. Rev. Phys. **2**(5), 229 (2020), doi:10.1038/s42254-020-0170-z.

[79] N. Goldman and J. Dalibard, *Periodically Driven Quantum Systems: Effective Hamiltonians and Engineered Gauge Fields*, Phys. Rev. X **4**, 031027 (2014), doi:10.1103/PhysRevX.4.031027.

[80] E. Sarrouy and J.-J. Sinou, *Non-Linear Periodic and Quasi-Periodic Vibrations in Mechanical Systems - On the use of the Harmonic Balance Methods*, Advances in Vibration Analysis Research **21**, 419 (2011), doi:10.5772/15638.

[81] *Multi-Dimensional Harmonic Balance Applied to Rotor Dynamics*, vol. Volume 1: 21st Biennial Conference on Mechanical Vibration and Noise, Parts A, B, and C of *International Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, doi:10.1115/DETC2007-34792 (2007).

[82] D. Cox, J. Little and D. O'Shea, *Ideals, varieties, and algorithms: an introduction to computational algebraic geometry and commutative algebra*, Springer, doi:10.1007/978-3-319-16721-3 (2013).

[83] D. J. Bates, A. J. Sommese, J. D. Hauenstein and C. W. Wampler, *Numerically solving polynomial systems with Bertini*, SIAM (2013).

[84] J. Verschelde, *Algorithm 795: PHCpack: A general-purpose solver for polynomial systems by homotopy continuation*, ACM Trans. Math. Softw. **25**(2), 251 (1999), doi:10.1145/317275.317286.

[85] M. Dykman, ed., *Fluctuating Nonlinear Oscillators*, Oxford University Press, doi:10.1093/acprof:oso/9780199691388.001.0001 (2012).

[86] *HarmonicBalance.jl documentation*, https://nonlinearoscillations.github.io/HarmonicBalance.jl.

[87] S. Gowda, Y. Ma, A. Cheli, M. Gwóźdź, V. B. Shah, A. Edelman and C. Rackauckas, *High-Performance Symbolic-Numerics via Multiple Dispatch*, ACM Commun. Comput. Algebra **55**(3), 92–96 (2022), doi:10.1145/3511528.3511535.

[88] J. D. Hunter, *Matplotlib: A 2D graphics environment*, Comput. Sci. Eng. **9**(3), 90 (2007), doi:10.1109/MCSE.2007.55.

[89] J. Verley, E. R. Keiter and H. K. Thornquist, *Xyce: Open Source Simulation for Large-Scale Circuits.*, Tech. rep., Sandia National Lab.(SNL-NM), Albuquerque, NM (United States) (2018).

[90] A. Halbach, *Sparselizard-the user friendly finite element c++ library* (2017).

[91] R. Veltz, *BifurcationKit.jl* (2020).

[92] R. Genesio and A. Tesi, *Harmonic balance methods for the analysis of chaotic dynamics in nonlinear systems*, Automatica **28**(3), 531 (1992), doi:https://doi.org/10.1016/0005-1098(92)90177-H.

[93] N. M. Krylov and N. N. Bogoliubov, *Introduction to Non-Linear Mechanics.(AM-11), Volume 11*, Princeton University Press (2016).

[94] S. Krämer, D. Plankensteiner, L. Ostermann and H. Ritsch, *QuantumOptics.jl: A Julia framework for simulating open quantum systems*, Comput. Phys. Commun. **227**, 109 (2018), doi:https://doi.org/10.1016/j.cpc.2018.02.004.

[95] J. F. <johannes.feist@gmail.com> and contributors, *QuantumAlgebra.jl*, doi:10.5281/zenodo.5094080 (2021).