Tor: a toolbox for the continuation of two-dimensional tori in autonomous systems and non-autonomous systems with periodic forcing

Mingwu Li *

Institute for Mechanical Systems, ETH Zürich, Switzerland

Abstract

We present a toolbox for the continuation of two-dimensional tori in autonomous dynamical systems and non-autonomous systems with periodic forcing. A torus is solved as the solution to a boundary-value problem. Truncated Fourier series and collocation methods are used to discretize the boundary-value problem. This toolbox is based on the continuation package COCO. A few examples are included to demonstrate the functionality of this toolbox.

1 Introduction

Consider autonomous dynamical system

$$\frac{dy}{dt} = f(y, p) \tag{1}$$

where $f : \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^n$ and non-autonomous system

$$\frac{dy}{dt} = f(t, y, p) \tag{2}$$

where $f : \mathbb{R} \times \mathbb{R}^n \times \mathbb{R}^q \to \mathbb{R}^n$ is periodic in t, namely, there exists some positive T such that $f(t, y, p) = f(t+T, y, p)$ for all t, y and p. The goal of this report is presenting a unified toolbox `tor`[1] for the continuation of *two-dimensional tori* in the above two types of dynamical systems.

This toolbox includes several core functions as follows

- `ode_TR2tor`: continuation of tori from a *Neimark-Sacker/torus* bifurcation periodic orbit,
- `ode_isol2tor`: continuation of tori from an initial solution guess,
- `ode_tor2tor`: continuation of tori from a saved torus solution,
- `ode_BP2tor`: continuation of tori along a secondary branch passing through a branch point.

This toolbox is based on the continuation package COCO [4]. There are examples of continuation of tori in COCO. Specifically, the continuation of tori in autonomous *Langford* dynamical system is studied in Chapter 9 of [2], a book documents the development the package in great detail. In the `coll` tutorial of COCO [1], the continuation of tori in a non-autonomous system with harmonic excitation is studied. Based on these examples, we develop a *general-purpose* toolbox for the continuation of tori in dynamical systems.

In the continuation of periodic orbits, Neimark-Sacker/torus (TR) bifurcation may be detected. It follows that a family of tori emanating from the bifurcation point can be obtained. In `po` toolbox of COCO, one can perform the continuation of periodic orbits. Once a TR bifurcation periodic solution is found using `po`,

*Email: mingwli@ethz.ch

[1] `tor` toolbox is available at `https://github.com/mingwu-li/torus_collocation`

one may switch to the continuation of tori starting from such a bifurcation periodic orbit using `ode_TR2tor`. Therefore, the `tor` toolbox connects to the `po` toolbox seamlessly.

The rest of this report is organized as follows. In section 2, problem formulation for tori is given. Specifically, a partial differential equation and phase conditions are derived to solve tori by boundary-value problem approach. Discretization schemes of the boundary-value problem are then discussed in Section 3. In section 4, parameter continuation of tori is discussed. Following the analysis of dimension deficit of the continuation problem, the initialization of torus solution is presented to construct initial torus solution born from a TR bifurcation periodic orbit. In section 5, several examples are presented to demonstrate the effectiveness of the toolbox. Finally, section 6 concludes this report with discussion on the limitation and future enhancement of the current toolbox.

2 Problem formulation

2.1 PDE formulation

In this section, we focus on the case of non-autonomous dynamical systems. It is straightforward to see that the same formulation holds for autonomous systems as well. The formulation has been derived in [2, 1]. We include the derivation here for completeness

For a 2-dimensional quasi-periodic invariant torus, we introduce a torus function $u : \mathbb{S}\times\mathbb{S}\to\mathbb{T}^2$ and two frequencies $\omega_1, \omega_2 \in \mathbb{R}$ such that the parallel flow

$$\frac{d\theta_i}{dt} = \omega_i, \quad i = 1, 2. \tag{3}$$

Substitution of $y(t) = u(\theta_1(t), \theta_2(t))$ into the vector field yields

$$\omega_1 \frac{\partial u}{\partial \theta_1}(\theta_1, \theta_2) + \omega_2 \frac{\partial u}{\partial \theta_2}(\theta_1, \theta_2) = f(t, u(\theta_1, \theta_2), p). \tag{4}$$

The above equation is a first order *quasi-linear* partial differential equation (PDE). With the method of characteristics, we let $\theta_1 = \varphi + \omega_1 t$ and $\theta_2 = \omega_2 t$ and introduce

$$v(\varphi, t) := u(\varphi + \omega_1 t, \omega_2 t) \tag{5}$$

for $\varphi \in \mathbb{S}$ and $t \in [0, 2\pi/\omega_2]$. From the transformation $(\varphi, t) \mapsto (\theta_1, \theta_2)$, we obtain $\varphi = \theta_1 - \varrho\theta_2$ and $t = \theta_2/\omega_2$ with $\varrho = \omega_1/\omega_2$. It follows that

$$\omega_1 \frac{\partial u}{\partial \theta_1}(\theta_1, \theta_2) = \omega_1 \left(\frac{\partial v}{\partial \varphi}\frac{\partial \varphi}{\partial \theta_1} + \frac{\partial v}{\partial t}\frac{\partial t}{\partial \theta_1} \right) = \omega_1 \frac{\partial v}{\partial \varphi}, \tag{6}$$

$$\omega_2 \frac{\partial u}{\partial \theta_2}(\theta_1, \theta_2) = \omega_2 \left(\frac{\partial v}{\partial \varphi}\frac{\partial \varphi}{\partial \theta_2} + \frac{\partial v}{\partial t}\frac{\partial t}{\partial \theta_2} \right) = -\omega_1 \frac{\partial v}{\partial \varphi} + \frac{\partial v}{\partial t}, \tag{7}$$

and the vector field becomes

$$\frac{\partial v}{\partial t}(\varphi, t) = f(t, v(\varphi, t), p), \quad \varphi \in \mathbb{S}, t \in [0, T] \tag{8}$$

where $T := 2\pi/\omega_2$.

As for boundary conditions, note that

$$v(\varphi, 0) = u(\varphi, 0), \tag{9}$$

$$u(\varphi + 2\pi\varrho, 0) = u(\varphi + 2\pi\varrho, 2\pi) = u(\varphi + \omega_1 \cdot 2\pi/\omega_2, \omega_2 \cdot 2\pi/\omega_2) = v(\varphi, 2\pi/\omega_2) = v(\varphi, T), \tag{10}$$

and then an *all-to-all* coupling condition is obtained as follows

$$v(\varphi + 2\pi\varrho, 0) = v(\varphi, T), \quad \forall \varphi \in \mathbb{S}. \tag{11}$$

We have a boundary-value problem (BVP) governed by (8) and (11). However, we need phase conditions to yield unique solution to the BVP.

2.2 Phase conditions

In the case of non-autonomous systems, we need a phase condition given $\varphi \in \mathbb{S}$. We use Poincaré phase condition as follows

$$\langle v_\varphi^*(0,0), v(0,0) - v^*(0,0)\rangle = 0, \tag{12}$$

where $v^*(\varphi,t)$ is a *known* function of φ and t, and v_φ^* is its partial derivative with respect to φ. During continuation, $v^*(\varphi,t)$ will be updated as the solution of previous continuation step and then the Poincaré section is *moving* during continuation. Integral phase condition provides an alternative to Poincaré phase condition. The reader may refer to [5] for more details about the integral phase condition.

In the case of autonomous systems, we need one more phase condition given (8) becomes autonomous. For consistency, we again use Poincaré phase condition

$$\langle v_t^*(0,0), v(0,0) - v^*(0,0)\rangle = 0, \tag{13}$$

where v_t^* is the partial derivative of $v^*(\varphi,t)$ with respect to t. Likewise, the Poincaré section here is updated during continuation.

3 Discretization

One need to discretize the domain $\mathbb{S}\times[0,T]$ to obtain approximated BVP. We first apply truncated Fourier series expansion of $v(\varphi,t)$ in φ, yielding a *multi-segments* BVP with ordinary differential equations (ODEs). Then, collocation method is used to approximate the obtained ODEs to yield a set of nonlinear algebraic equations.

3.1 Fourier series expansion

Following [2], let χ represent truncated Fourier series expansion of a component of $v(\varphi,\tau)$, we have

$$v_i(\varphi,t) \approx \chi(\varphi,t) := a_0(t) + \sum_{k=1}^{N} a_k(t)\cos(k\varphi) + b_k(t)\sin(k\varphi). \tag{14}$$

It follows that

$$\begin{pmatrix} \chi(0,t) \\ \chi(\frac{2\pi}{2N+1},t) \\ \vdots \\ \chi(\frac{4\pi N}{2N+1},t) \end{pmatrix} = \mathcal{F}^{-1} \begin{pmatrix} a_0(t) \\ a_1(t) \\ b_1(t) \\ \vdots \\ a_N(t) \\ b_N(t) \end{pmatrix} \tag{15}$$

where $\mathcal{F}$ is *discrete Fourier transform matrix* and given by equation 9.21 in [2].

Likewise, we have

$$v_i(\varphi+2\pi\varrho,t) \approx \chi(\varphi+2\pi\varrho,t) := a_0'(t) + \sum_{k=1}^{N} a_k'(t)\cos(k\varphi) + b_k'(t)\sin(k\varphi). \tag{16}$$

It follows that

$$\begin{pmatrix} a_0'(t) \\ a_1'(t) \\ b_1'(t) \\ \vdots \\ a_N'(t) \\ b_N'(\tau) \end{pmatrix} = \mathcal{R} \begin{pmatrix} a_0(t) \\ a_1(t) \\ b_1(t) \\ \vdots \\ a_N(t) \\ b_N(t) \end{pmatrix} \tag{17}$$

where $\mathcal{R}$ is given by equation (38) in [1] or equation (9.24) in [2].

From the derived *all-to-all* coupling boundary condition, we have

$$\chi(\varphi, T) = \chi(\varphi + 2\pi\varrho, 0), \tag{18}$$

or equivalently

$$a_0(T) + \sum_{k=1}^{N} a_k(T)\cos(k\varphi) + b_k(T)\sin(k\varphi) = a_0'(0) + \sum_{k=1}^{N} a_k'(0)\cos(k\varphi) + b_k'(0)\sin(k\varphi) \tag{19}$$

from which we obtain

$$\begin{pmatrix} a_0'(0) \\ a_1'(0) \\ b_1'(0) \\ \vdots \\ a_N'(0) \\ b_N'(0) \end{pmatrix} = \begin{pmatrix} a_0(T) \\ a_1(T) \\ b_1(T) \\ \vdots \\ a_N(T) \\ b_N(T) \end{pmatrix}. \tag{20}$$

Substitution yields

$$\mathcal{R}\begin{pmatrix} a_0(0) \\ a_1(0) \\ b_1(0) \\ \vdots \\ a_N(0) \\ b_N(0) \end{pmatrix} = \begin{pmatrix} a_0(T) \\ a_1(T) \\ b_1(T) \\ \vdots \\ a_N(T) \\ b_N(T) \end{pmatrix} \implies \mathcal{R}\mathcal{F}\begin{pmatrix} \chi(0,0) \\ \chi(\frac{2\pi}{2N+1},0) \\ \vdots \\ \chi(\frac{4\pi N}{2N+1},0) \end{pmatrix} = \mathcal{F}\begin{pmatrix} \chi(0,T) \\ \chi(\frac{2\pi}{2N+1},T) \\ \vdots \\ \chi(\frac{4\pi N}{2N+1},T) \end{pmatrix}. \tag{21}$$

We proceed to discretize the continuous family $v(\varphi, \tau)$ by restricting attention to the mesh $\{\varphi_j\}_{j=1}^{2N+1}$, where $\varphi_j := \frac{2(j-1)\pi}{2N+1}$. Following the above equation, the discrete *all-to-all* boundary condition is given by

$$(\mathcal{F} \otimes I_n)\begin{pmatrix} v(\varphi_1, T) \\ v(\varphi_2, T) \\ \vdots \\ v(\varphi_N, T) \end{pmatrix} = ((\mathcal{R}\mathcal{F}) \otimes I_n)\begin{pmatrix} v(\varphi_1, 0) \\ v(\varphi_2, 0) \\ \vdots \\ v(\varphi_N, 0) \end{pmatrix}, \tag{22}$$

where I_n denotes identity matrix of size n.

In addition, we obtain the following set of ODEs

$$\frac{dv(\varphi_j, t)}{dt} = f(t, v(\varphi_j, t), p), \quad j = 1, \cdots, 2N+1, t \in [0, T]. \tag{23}$$

Along with the boundary condition (22) and appropriate discrete phase conditions (which will be discussed in section 3.3), we have a *multi-segments* BVP with a set of ODEs.

3.2 Collocation

We apply collocation methods to solve the above multi-segments BVP. Here we summarize the basic idea of collocation methods. One may refer to [2] for more details.

1. Division of domain: the domain $[0, T]$ is divided into subintervals.
2. Approximation of unknown functions: the unknown function within each subinterval is approximated using Lagrangian interpolation of values of the function at a set of base points, and the continuity of the function between two adjacent subintervals is imposed.
3. Approximation of differential equations: the ODEs are satisfied at a set of collocation nodes.

The collocation method with adaptive mesh has been implemented in `coll` toolbox of COCO. The `tor` toolbox here is built upon the `coll` toolbox.

3.3 Phase conditions

The imposition of phase conditions (12) and (13) is straightforward with the above discretization. We have $v_t^*(0,0)=f(0,v^*(0,0),p)$. To obtain the approximation of $v_\varphi^*(0,0)$, let recall (14) and then

$$\chi_\varphi(\varphi,t)=\sum_{k=1}^{N}(kb_k(t))\cos(k\varphi)+(-ka_k(t))\sin(k\varphi). \tag{24}$$

Plugging $(\varphi,t)=(0,0)$ yields

$$\chi_\varphi(0,0)=\sum_{k=1}^{N}kb_k(0). \tag{25}$$

Recall

$$\begin{pmatrix}a_0(0)\\ a_1(0)\\ b_1(0)\\ \vdots\\ a_N(0)\\ b_N(0)\end{pmatrix}=\mathcal{F}\begin{pmatrix}\chi(\varphi_1,0)\\ \chi(\varphi_2,0)\\ \vdots\\ \chi(\varphi_{2N+1},0)\end{pmatrix}, \tag{26}$$

we can express $\chi_\varphi(0,0)$ as a linear function of $\{\chi(\varphi_j,0)\}_{j=1}^{2N+1}$. It follows that we can obtain $v_\varphi^*(0,0)$ in terms of $\{v^*(\varphi_j,0)\}_{j=1}^{2N+1}$.

4 Parameter continuation

4.1 Dimension deficit

Recall system parameters $p\in\mathbb{R}^q$, we regard ω_1, ω_2 and ϱ as system parameters as well. Other than the *all-to-all* coupling condition, we have the following boundary conditions

$$T_0=0,T=\frac{2\pi}{\omega_2},\varrho=\frac{\omega_1}{\omega_2}, \tag{27}$$

where T_0 is the initial time. With initially inactive system parameters, the dimension deficit is -1. We also need to account for phase conditions. It follows that

- *autonomous systems*: two phase conditions are included and then the dimension deficit is -3. So four parameters need to be released to yield an one-dimensional solution manifold. For instance, one may release ω_1,ω_2 and $p_{1,2}$ to obtain a family of tori with fixed rotation number ϱ,
- *non-autonomous systems*: one phase condition is included and we have a coupling condition for system parameters, namely, $\Omega_2-\omega_2=0$, where Ω_2 is a parameter included in p and characterizes the period of non-autonomous forcing ($T=2\pi/\Omega_2$). So the dimension deficit is again -3 and we need to release four parameters to obtain a one-dimensional manifold of tori.

4.2 Initialization

In the continuation of periodic orbits, Neimark-Sacker (torus) bifurcation may be observed. The occurrence of torus bifurcation indicates the born of tori. Here we provide an initial guess to such a torus. The function `ode_TR2tor` is based on the initialization presented in this section. The initialization here is adapted from the derivation in [3].

Consider dynamical system

$$\dot{x}=f(t,x,p),\quad x(t_0)=x_0 \tag{28}$$

with a solution $x(t)$. It follows that

$$x(t)=x_0+\int_{t_0}^{t}f(s,x(s),p)\mathrm{d}s \tag{29}$$

and hence

$$\frac{\partial x(t)}{\partial x_0} = \mathbb{I} + \int_{t_0}^{t} \frac{\partial f(s, x(s), p)}{\partial x} \frac{\partial x(s)}{\partial x_0} \mathrm{d}s. \tag{30}$$

Let $\Phi(t, t_0) := D_{x_0} x(t) = \frac{\partial x(t)}{\partial x_0}$, we have

$$\dot{\Phi} = f_x \Phi, \quad \Phi(t_0) = \mathbb{I} \tag{31}$$

and the transition matrix $\mathcal{M}(t, t_0)$ is given by $\Phi(t, t_0)$ and the monodromy matrix is $\mathcal{M}(t_0 + T, t_0)$. In general, we solve the above variational equation with initial condition Φ_0. Let $\delta x_0 = \Phi_0 v$, it follows that

$$\delta x(t) = \Phi(t, t_0) v = \Phi(t, t_0) \Phi_0^{-1} \delta x_0 \implies \mathcal{M}(t, t_0) = \Phi(t, t_0) \Phi_0^{-1}. \tag{32}$$

Suppose $\mathcal{M}(t_0 + T, t_0)$ has an eigenvalue $e^{\mathrm{i}\alpha}$ and an eigenvector v such that

$$\mathcal{M}(t_0 + T, t_0) v = e^{\mathrm{i}\alpha} v. \tag{33}$$

$\mathrm{Re}(v)$ and $\mathrm{Im}(v)$ span an invariant subspace to the monodromy matrix. Define $u(t) = e^{-\mathrm{i}\alpha(t-t_0)/T} \mathcal{M}(t, t_0) v$, it follows that $u(t_0 + T) = u(t_0) = v$ and then $u(t)$ is a periodic function with period T. For a periodic reference solution $x_p(t)$ with period T, we know the coefficient matrix f_x is time-periodic and then the transition matrix $\mathcal{M}(t, t_0)$ satisfies

$$\mathcal{M}(t+T, t_0) = \mathcal{M}(t+T, t)\mathcal{M}(t, t_0) = \mathcal{M}(t+T, t_0+T)\mathcal{M}(t_0+T, t_0) = \mathcal{M}(t, t_0)\mathcal{M}(t_0+T, t_0) \tag{34}$$

where the first two equalities use the semi-group property of state transition matrix and the last one uses the periodicity of coefficient matrix. Then we can show that $u(t)$ is an eigenvector to $\mathcal{M}(t+T, t)$ with same eigenvalue $e^{\mathrm{i}\alpha}$. Note

$$\mathcal{M}(t+T, t)\mathcal{M}(t, t_0) v = \mathcal{M}(t, t_0)\mathcal{M}(t_0+T, t_0) v = e^{\mathrm{i}\alpha} \mathcal{M}(t, t_0) v, \tag{35}$$

we have $\mathcal{M}(t+T, t) u(t) = e^{\mathrm{i}\alpha} u(t)$.

So we can define a torus function perturbation

$$\hat{u}(\theta_1, t) = \mathrm{Re}[e^{\mathrm{i}\theta_1} u(t)] = \cos\theta_1 \mathrm{Re}(u(t)) - \sin\theta_1 \mathrm{Im}(u(t)), \quad \theta_1 \in \mathbb{S}. \tag{36}$$

It follows that

$$\mathcal{M}(t+T)\hat{u}(\theta_1, t) = \mathrm{Re}[\mathcal{M}(t+T) e^{\mathrm{i}\theta_1} u(t)] = \mathrm{Re}[e^{\mathrm{i}\theta_1} \mathcal{M}(t+T) u(t)] = \mathrm{Re}[e^{\mathrm{i}\theta_1} e^{\mathrm{i}\alpha} u(t)] = \hat{u}(\theta_1 + \alpha, t) \tag{37}$$

which indicates the rotation of $\hat{u}(\theta_1, t)$ with angle α when state transition is applied. The perturbed torus solution is then given by

$$\tilde{u}(\theta_1, t) = \mathcal{M}(t, t_0)\hat{u}(\theta_1, t_0) = \mathrm{Re}[e^{\mathrm{i}\theta_1} \mathcal{M}(t, t_0) v] = \hat{u}(\theta_1 + \alpha(t - t_0)/T, t). \tag{38}$$

So the torus initial solution is given by

$$\hat{x}(\theta_1, t) = x(t) + \epsilon \tilde{u}(\theta_1, t) = x(t) + \epsilon \hat{u}(\theta_1 + \omega_1(t - t_0), t), \quad \forall \theta_1 \in \{\varphi_j\}_{j=1}^{2N+1} \tag{39}$$

where ϵ controls the amount of perturbation. We have used the fact that $T = \frac{2\pi}{\omega_2}$ and $\alpha = \omega_1 T$.

Note that the eigenvector is still an eigenvector after multiplication by a complex number. So we would like to choose the one such that $\langle v_{\mathrm{R}}, v_{\mathrm{I}} \rangle = 0$, where $v_{\mathrm{R}} = \mathrm{Re}(v)$ and $v_{\mathrm{I}} = \mathrm{Im}(v)$. Let the complex number be $e^{\mathrm{i}\theta}$, it follows that

$$e^{\mathrm{i}\theta}(v_{\mathrm{R}} + \mathrm{i}v_{\mathrm{I}}) = (\cos\theta + \mathrm{i}\sin\theta)(v_{\mathrm{R}} + \mathrm{i}v_{\mathrm{I}}) = (\cos\theta v_{\mathrm{R}} - \sin\theta v_{\mathrm{I}}) + \mathrm{i}(\cos\theta v_{\mathrm{I}} + \sin\theta v_{\mathrm{R}}). \tag{40}$$

We ask for

$$\langle \cos\theta v_{\mathrm{R}} - \sin\theta v_{\mathrm{I}}, \cos\theta v_{\mathrm{I}} + \sin\theta v_{\mathrm{R}} \rangle = 0, \tag{41}$$

from which we obtain

$$\langle v_{\mathrm{R}}, v_{\mathrm{I}} \rangle \cos 2\theta + 0.5(\langle v_{\mathrm{R}}, v_{\mathrm{R}} \rangle - \langle v_{\mathrm{I}}, v_{\mathrm{I}} \rangle) \sin 2\theta = 0 \implies \tan 2\theta = \frac{2\langle v_{\mathrm{R}}, v_{\mathrm{I}} \rangle}{\langle v_{\mathrm{I}}, v_{\mathrm{I}} \rangle - \langle v_{\mathrm{R}}, v_{\mathrm{R}} \rangle}. \tag{42}$$

We can solve θ from the above equation with `atan2` function.

5 Examples

5.1 Langford dynamical system

Consider the *Langford* dynamical system

$$\dot{x}_1 = (x_3 - 0.7)x_1 - \omega x_2, \tag{43}$$

$$\dot{x}_2 = \omega x_1 + (x_3 - 0.7)x_2, \tag{44}$$

$$\dot{x}_3 = 0.6 + x_3 - \frac{1}{3}x_3^3 - (x_1^2 + x_2^2)(1 + \rho x_3) + \epsilon x_3 x_1^3. \tag{45}$$

Following the analysis in [2], the system has a family of periodic orbits when $\epsilon = 0$, and such solution undergoes a torus bifurcation at $\rho = \rho^* \approx 0.615$.

We proceed to use `po` toolbox in COCO to find the family of periodic orbits and detect TR bifurcation point along the solution manifold. Specifically, we first use `ode45` to perform forward simulation and generate an approximated periodic orbit solution. Then we call `ode_isol2po` to encode the boundary-value problem of periodic orbits.

```
%% construct initial periodic orbit
p0 = [3.5; 1.5; 0]; % [om ro eps]
T  = 2*pi/p0(1);
[~,x0] = ode45(@(t,x) lang(x,p0), 0:100*T, [0.3; 0.4; 0]); % transient
[t0,x0] = ode45(@(t,x) lang(x,p0), linspace(0,T,100), x0(end,:)); % periodic solution
figure;
plot(t0,x0);

%% continuation of periodic orbit
prob = coco_prob();
prob = ode_isol2po(prob, '', @lang, @lang_DFDX, @lang_DFDP, ...
  t0, x0, 'om','rho','eps', p0);

coco(prob, 'po', [], 1, 'rho', [0.2 2]);
```

Here `lang` denotes the vector field of the dynamical system, `lang_DFDX` and `lang_DFDP` represent the derivative of the vector field with respect to state x and system parameters p respectively. The system parameters in this example are (ω, ρ, ϵ) and named as `om`, `rho` and `eps` as above. The vector field and its derivatives are encoded as follows

```
function y = lang(x, p)

x1  = x(1,:);
x2  = x(2,:);
x3  = x(3,:);
om  = p(1,:);
ro  = p(2,:);
eps = p(3,:);

y(1,:) = (x3-0.7).*x1-om.*x2;
y(2,:) = om.*x1+(x3-0.7).*x2;
y(3,:) = 0.6+x3-x3.^3/3-(x1.^2+x2.^2).*(1+ro.*x3)+eps.*x3.*x1.^3;

end

function J = lang_DFDX(x, p)

x1  = x(1,:);
```

```
x2  = x(2,:);
x3  = x(3,:);
om  = p(1,:);
ro  = p(2,:);
eps = p(3,:);

J = zeros(3,3,numel(x1));
J(1,1,:) = (x3-0.7);
J(1,2,:) = -om;
J(1,3,:) = x1;
J(2,1,:) = om;
J(2,2,:) = (x3-0.7);
J(2,3,:) = x2;
J(3,1,:) = -2*x1.*(1+ro.*x3)+3*eps.*x3.*x1.^2;
J(3,2,:) = -2*x2.*(1+ro.*x3);
J(3,3,:) = 1-x3.^2-ro.*(x1.^2+x2.^2)+eps.*x1.^3;

end

function J = lang_DFDP(x, p)

x1 = x(1,:);
x2 = x(2,:);
x3 = x(3,:);

J = zeros(3,size(p,1),numel(x1));
J(1,1,:) = -x2;
J(2,1,:) = x1;
J(3,2,:) = -x3.*(x1.^2+x2.^2);
J(3,3,:) = x3.*x1.^3;

end
```

Indeed, a TR bifurcation periodic orbit is found at $\rho \approx 0.61545$. We then move to the continuation of tori in the system. We call `ode_TR2tor` to switch from the continuation of periodic orbits to the continuation of tori.

```
T_po   = 5.3;
T_ret  = 2*pi/3.5;
varrho = T_ret/T_po;
bd     = coco_bd_read('po');
TRlab = coco_bd_labs(bd, 'TR');
prob = coco_prob();
prob = coco_set(prob, 'cont', 'NAdapt', 5, 'h_min',...
    1e-3, 'PtMX', 50, 'h_max', 10, 'bi_direct', false);
prob = ode_TR2tor(prob, '', 'po', TRlab, 50);

coco(prob, 'tr1', [], 1, 'varrho','rho','om1','om2','eps',[varrho,0.44]);
```

Here `varrho`, `om1` and `om2` correspond to ϱ, ω_1 and ω_2 in the PDE formulation respectively. `TRlab` gives the label of TR bifurcation solution in `po` run. The 50 in `prob = ode_TR2tor(prob, '', 'po', TRlab, 50)` characterizes the number of segments, namely, N in section 3. Recall the number of segments is given by $2N+1$, so we have 101 segments in this case. By default, $N = 10$. Please type `help ode_TR2tor` in MATLAB for more details of the syntax of this constructor.

In the above continuation run `tr1`, $(\varrho, \rho, \omega_1, \omega_2)$ are varied and a one-dimensional solution manifold is obtained. As discussed in section 4.1, four parameters need to be released to yield a one-dimensional manifold.

Indeed, eps does not change during the continuation run. As an alternative, we may fix ϱ and then another one-dimensional manifold of tori is obtained. To demonstrate it, we call constructor ode_tor2tor as follows

```
%% continuation of torus from previous solution
bd   = coco_bd_read('tr1');
lab  = coco_bd_labs(bd, 'EP');
lab  = max(lab);
prob = coco_prob();
prob = coco_set(prob, 'cont', 'NAdapt', 5, 'h_min',...
    1e-3, 'PtMX', 30, 'h_max', 10, 'bi_direct', true);
prob = ode_tor2tor(prob, '', 'tr1', lab);

coco(prob, 'tr2', [], 1, 'eps','rho','om1','om2','varrho');
```

Here we start continuation from a solution found in previous run tr1. In current run tr2, ϱ is not varied in continuation, as can be seen in the continuation history.

Once a torus solution is found, you may call plot_torus to visualize the solution. When ρ is away from ρ^*, the size of tori born from TR periodic orbit grows, as can be seen in the visualization of a family of tori

```
%% visualization
figure; coco_plot_bd('tr2','rho','eps');
for lab = 1:5
    plot_torus('','tr1', lab, [1 2 3], 0.75); pause(1);
end
```

To validate an obtained torus solution, one may perform forward simulation with an initial condition on the torus

```
lab = 5;
plot_torus('','tr1', lab, [1 2 3]);hold on
sol = tor_read_solution('','tr1',lab);
p   = sol.p(1:end-3);
xbp = sol.xbp;
[t,x] = ode45(@(t,x) lang(x,p), 0:0.01:100*T, xbp(1,:,1));
plot3(x(:,1),x(:,2),x(:,3),'r-');
```

Here tor_read_solution extracts the result of continuation run. For more details of the arguments of plot_torus and tor_read_solution, please check their corresponding help info.

5.2 Van der Pol oscillator

Consider Van der Pol oscillator subject to harmonic excitation

$$\ddot{x} - c(1-x^2)\dot{x} + x = a\cos\Omega t. \tag{46}$$

We are interested in the quasiperiodic response of such a system.

We start by constructing initial solution guess with forward simulation

```
% pnames = [  om     c     a ]
p0       = [ 1.5111; 0.11; 0.1 ];

T_po = 2*pi; % Approximate period
N    = 10;   % 2N+1 = Number of orbit segments
tout = linspace(0, T_po, 2*N+2);

T_ret = 2*pi/p0(1); % return time = 2*pi/om
tt    = linspace(0,1,10*(2*N+1))';
```

```
t1    = T_ret*tt;
x0    = zeros(numel(tt),2,2*N+1);

figure; hold on
for i=1:2*N+1
% For each point on orbit, flow for return time and reconstitute 3D trajectory
  x1       = [2*cos(tout(i)) 2*sin(tout(i))];
  [~, x1] = ode45(@(t,x) vdp(t,x,p0), 10*t1, x1); % Transient simulation
  [~, x1] = ode45(@(t,x) vdp(t,x,p0), t1, x1(end,:));
  x0(:,:,i) = x1;
  plot3(x1(:,1),t1,x1(:,2),'r-');
end

varrho = T_ret/T_po;
```

We proceed to use constructor `ode_isol2tor` to perform continuation of tori with the generated initial solution guess. Given the dynamical system is non-autonomous, `coco_set` is called to change default *autonomous* setting. In addition, the excitation frequency parameter Ω needs to be named as `Om2` given the coupling condition $\Omega_2 - \omega_2$ needs to be imposed, as discussed in section 4.1. With these observations, we have

```
prob = coco_prob();
prob = coco_set(prob, 'tor', 'autonomous', false);
prob = coco_set(prob, 'coll', 'NTST', 40);
prob = coco_set(prob, 'cont', 'NAdapt', 0, 'h_max', 2, 'PtMX', 60);
torargs = {@vdp @vdp_DFDX @vdp_DFDP @vdp_DFDT t1 x0 'Om2','c','a','om1','om2','varrho'
    [p0' -2*pi/T_po p0(1) -varrho]};

prob = ode_isol2tor(prob, '', torargs{:});

coco(prob, 'vdP_torus', [], 1, 'a','Om2','om2','om1', 'varrho','c', {[0.1 2] [1 2]});
```

Here we choose negative ω_1 and ϱ because the rotation direction could be in opposite direction. In other words, the value of ϱ can be both positive and negative. One may predict that ϱ is not changed in the continuation run `vdP_torus` because releasing the first four parameters $(a, \Omega_2, \omega_2, \omega_1)$ is enough to yield a one-dimensional solution manifold of tori. Indeed ϱ does not change in this continuation run.

Next we use constructor `ode_tor2tor` to perform continuation from a saved torus solution in previous run `vdP_torus`.

```
bd  = coco_bd_read('vdP_torus');
lab = coco_bd_labs(bd, 'EP');
lab = max(lab);
prob = coco_prob();
prob = coco_set(prob, 'cont', 'NAdapt', 5, 'h_max', 2, 'PtMX', 60);
prob = ode_tor2tor(prob, '', 'vdP_torus', lab);

coco(prob, 'vdP_torus_varrho', [], 1,
'a','Om2','om2','varrho','om1','c', {[0.1 2] [1 2]});
```

In the above continuation run `vdP_torus_varrho`, ϱ is free to change, and several branch points are detected. We then proceed to switch to secondary solution branch passing through one branch point. We use constructor `ode_BP2tor` to perform such switch.

```
bd  = coco_bd_read('vdP_torus_varrho');
lab = coco_bd_labs(bd, 'BP');
lab = lab(1);
```

```
prob = coco_prob();
prob = coco_set(prob, 'cont', 'NAdapt', 0, 'h_max', 2, 'PtMX', 60);
prob = ode_BP2tor(prob, '', 'vdP_torus_varrho', lab);

coco(prob, 'vdP_torus_varrho_BP', [], 1,
'a','Om2','om2','varrho','om1','c', [0.1 2] [1 2]);
```

As exercises to this example, the reader may follow the methodology of previous example. Specifically, continuation of periodic orbits is performed first and then switch to the continuation of tori by `ode_TR2tor` and detected TR points in the continuation run of periodic orbits. In addition, the reader may consider the case of positive ϱ and compare the results in two cases.

6 Discussion

A unified toolbox `tor` is presented in this report. There are some limitations of the discretization schemes used in the toolbox. First, the number of segments has to be determined before continuation and cannot be changed during continuation. The user needs to check the fidelity of obtained results and then update the number of segments if necessary. An error estimator is instructive for choosing reasonable number of segments. Second, the collocation method used here results in heavy computation load for large dynamical systems. As an alternative, parallel shooting/forward simulation may be utilized to remove such a bottleneck. Finally, the stability of torus solution is not available in current toolbox. Variational equation of torus may be derived and used to determine the stability of torus solution.

References

[1] Harry Dankowicz and Frank Schilder. The trajectory collocation toolbox.

[2] Harry Dankowicz and Frank Schilder. *Recipes for continuation*. SIAM, 2013.

[3] Zubin P Olikara and Kathleen C Howell. Computation of quasi-periodic invariant tori in the restricted three-body problem. *Paper No. AAS*, pages 1–15, 2010.

[4] Frank Schilder and Harry Dankowicz. Continuation core and toolboxes (COCO). `http://sourceforge.net/projects/cocotools`.

[5] Frank Schilder, Hinke M Osinga, and Werner Vogt. Continuation of quasi-periodic invariant tori. *SIAM Journal on Applied Dynamical Systems*, 4(3):459–488, 2005.