Nonlinear Parameter Continuation with COCO

Lecture given during
Advanced Summer School on
Continuation Methods for Nonlinear Problems

Harry Dankowicz

Department of Mechanical Science and Engineering
University of Illinois at Urbana-Champaign

August 13-24, 2018

Outline

The Collocation Continuation Problem

For continuation of approximate solutions of the dynamical system

$$\dot{x} = F(t, x, p),\ (x, p) \in \mathbb{R}^n \times \mathbb{R}^q,\ t \in [T_0, T_0 + T],$$

define

$$\Phi : (\upsilon, T_0, T, p) \mapsto \begin{pmatrix} \frac{T}{2N}\ \mathfrak{vec}\left(\kappa_F * F_{cn}\right) - W' \cdot \upsilon \\ Q \cdot \upsilon_{bp} \end{pmatrix}$$

in terms of the column matrix υ of unknown values of the state variables on a mesh of $N(m+1)$ *base points*, and the values

$$F_{cn} = F\left(T0 + Tt_{cn}, \mathfrak{vec}_n\left(W \cdot \upsilon\right), 1_{1,Nm} \otimes p\right)$$

of the vector field evaluated on a set t_{cn} of Nm *collocation nodes* on the interval $[0, 1]$.

The Collocation Continuation Problem

The dimensional deficit of the collocation zero problem equals

$$n + q + 2.$$

For an autonomous vector field, append the monitor function

$$u \mapsto T_0$$

and assign the index of the corresponding continuation parameter to $\mathbb{I}$. The dimensional deficit of the corresponding restricted continuation problem then equals

$$n + q + 1.$$

The Catenary Functional

In COCO, we construct an empty continuation problem using the `coco_prob` command:

```
>> prob = coco_prob();
```

The commands

```
>> cat = @(x,p) [x(2,:); (1+x(2,:).^2)./x(1,:)];
>> t0 = 0;
>> x0 = [1 0];
>> prob = ode_isol2coll(prob, '', cat, t0, x0, []);
```

append the collocation zero problem on a default mesh consisting of 10 intervals with 5 base points and 4 collocation nodes in each interval, associate T_0 with the inactive continuation parameter `'coll.T0'`, and initialize the continuation problem with

$$v = \mathfrak{vec}\left(1_{1,50} \otimes \begin{pmatrix} 1 & 0 \end{pmatrix}\right),\ T_0 = 0,\ T = 0,\ p = \emptyset.$$

The Catenary Functional

The commands

```
>> data = coco_get_func_data(prob, 'coll', 'data');
>> maps = data.coll_seg.maps;
>> prob = coco_add_pars(prob, 'pars', ...
     [ maps.x0_idx; maps.x1_idx(1); maps.T_idx ], ...
     { 'y1s' 'y2s' 'y1e' 'T' });
```

append the monitor functions

$$u \mapsto \begin{pmatrix} \upsilon_i \\ \upsilon_{f,1} \\ T \end{pmatrix},$$

label the corresponding continuation parameters by `'y1s'`, `'y2s'`, `'y1e'`, and `'T'`, and assign the corresponding indexes to $\mathbb{I}$.

The Catenary Functional

The dimensional deficit of the restricted continuation problem is now -1. The commands

```
>> cont_args = { 1, { 'T' 'y1e'  }, [0 1] };
>> coco(prob, 'coll1', [], cont_args{:});
```

identify the desired manifold dimension as 1, reassign the indexes of the continuation parameters `'T'` and `'y1e'` to $\mathbb{J}$, and restrict continuation to the domain `'T'`$\in [0, 1]$.

By default, the collocation continuation problem is non-adaptive, so the number and meaning of the continuation variables and the zero functions is unchanged during continuation.

The Catenary Functional

As an alternative, the commands

```
>> prob = coco_set(prob, 'cont', 'NAdapt', 10);
>> cont_args = { 1, { 'T' 'y1e'  }, [0 1] };
>> coco(prob, 'coll1', [], cont_args{:});
```

instruct the continuation algorithm to make adaptive changes to the problem discretization after every ten successive steps of continuation.

Such adaptive changes are designed to ensure that a suitably estimated discretization error remains below a critical threshold during continuation.

More frequent changes, or a finer initial mesh, may be required in order to ensure successful continuation across the desired computational domain.

The Bratu Problem

Consider the autonomous two-point boundary-value problem

$$\dot{x}_1 = x_2,\ \dot{x}_2 = -pe^{x_1},\ x_1(0) = 0,\ x_1(1) = 0$$

in terms of the vector of state variables $x = (x_1, x_2) \in \mathbb{R}^2$ and the scalar problem parameter $p \in \mathbb{R}$. When $p = 0$,

$$x_1(t) = x_2(t) = 0$$

is a trivial solution.

We encode the vector field and the boundary conditions in the anonymous functions `brat` and `brat_bc`.

```
>> brat    = @(x,p) [x(2,:); -p(1,:).*exp(x(1,:))];
>> brat_bc = @(~,T,x0,x1,p) [T-1; x0(1); x1(1)];
```

The Bratu Problem

The commands

```
>> coll_args = { brat, [0;1], zeros(2), 0 };
>> bvp_args  = [ coll_args, 'p', { brat_bc } ];
>> prob      = ode_isol2bvp(coco_prob(), bvp_args{:});
```

- construct the collocation zero problem on a default mesh consisting of 10 intervals with 5 base points and 4 collocation nodes in each interval,
- associate p and T_0 with inactive continuation parameters `'p'` and `'coll.T0'`,
- initialize the continuation problem with

$$v = \mathfrak{vec}\left(1_{1,50} \otimes \begin{pmatrix} 0 & 0 \end{pmatrix}\right),\ T_0 = 0,\ T = 1,\ p = 0,$$

- and append the boundary conditions

$$T = 1,\ v_{i,1} = v_{f,1} = 0.$$

The Bratu Problem

The dimensional deficit of the restricted continuation problem now equals 0. The command

```
>> bd = coco(prob, 'brat1', [], 1, 'p', [0 4]);
```

identifies the desired manifold dimension as 1, reassigns the index of the continuation parameter `'p'` to $\mathbb{J}$, and restricts continuation to the domain `'p'`$\in [0, 4]$.

Solutions exist for $p > 0$ provided that

$$p = \frac{4C^2}{1 + \cosh C}$$

for some C. This equation has two roots provided that $0 \leq p < p^* \approx 3.1583$ and no roots for $p > p^*$.

An Invariant Torus

Consider the non-autonomous dynamical system

$$\dot{\rho} = \rho\big(1 + \rho(\cos \omega t - 1)\big), \; \dot{\psi} = \Omega$$

in the polar coordinates ρ and ψ. Then, for $t \gg 1$,

$$\rho(t) \approx \rho^*(t) = \frac{1+\omega^2}{1+\omega^2 - \cos \omega t - \omega \sin \omega t}, \; \psi(t) = \Omega t + \psi_0$$

corresponding to motion on an invariant two-dimensional torus $\mathbb{T}$ described by the torus function

$$u : (\theta_1, \theta_2) \mapsto \big(\rho^*(\theta_2/\omega) \cos \theta_1, \rho^*(\theta_2/\omega) \sin \theta_1\big),$$

where $\dot{\theta}_1 = \Omega$ and $\dot{\theta}_2 = \omega$.

An Invariant Torus

The torus dynamics consist of either i) torus-covering trajectories, or ii) a continuous family of periodic orbits.

The definition $v(\varphi, \tau) := u(\varphi + \Omega\tau, \omega\tau)$ implies that

$$v(\varphi, 0) = u(\varphi, 0),\ v(\varphi, 2\pi/\omega) = u(\varphi + 2\pi\Omega/\omega, 0)$$

and

$$\frac{\partial v}{\partial \tau} = F(\tau, v(\varphi, \tau), p).$$

in terms of the original vector field F in cartesian coordinates.

We approximate the torus in terms of a sequence

$$\{2\pi(j-1)/(2M+1)\}_{j=1}^{2M+1}$$

of values of φ, a corresponding discretized approximation of $u(\varphi, 0)$, and a family of admissible trajectory segments $v(\varphi, \tau)$.

An Invariant Torus

Suppose that the boundary conditions take the form

$$\left(\mathcal{F} \otimes I_2\right) \cdot \begin{pmatrix} v\left(\varphi_1, 2\pi/\omega\right) \\ \vdots \\ v\left(\varphi_{2M+1}, 2\pi/\omega\right) \end{pmatrix} = \left(\left(\mathcal{R} \cdot \mathcal{F}\right) \otimes I_2\right) \cdot \begin{pmatrix} v\left(\varphi_1, 0\right) \\ \vdots \\ v\left(\varphi_{2M+1}, 0\right) \end{pmatrix}$$

in terms of the discrete Fourier transform matrix $\mathcal{F}$ and the rotation matrix $\mathcal{R}$ associated with a fixed rotation number Ω/ω.

Moreover, impose the phase condition $v_2(\varphi_1, 0) = 0$ to eliminate degeneracy associated with arbitrary shifts in φ.

An Invariant Torus

In terms of a collocation discretization, the corresponding *all-to-all*, multi-point boundary conditions read

$$\left(\mathcal{F}\otimes I_2\right)\cdot\begin{pmatrix} v_f^{(1)} \\ \vdots \\ v_f^{(2M+1)} \end{pmatrix} = \left(\left(\mathcal{R}\cdot\mathcal{F}\right)\otimes I_2\right)\cdot\begin{pmatrix} v_i^{(1)} \\ \vdots \\ v_i^{(2M+1)} \end{pmatrix},$$

$$T_0^{(1)} = \cdots = T_0^{(2M+1)} = 0,$$

$$T^{(1)} = \cdots = T^{(2M+1)} = 2\pi/\omega,$$

and

$$v_{i,2}^{(1)} = 0$$

resulting in a total dimensional deficit equal to 1.