

Collocation for CBC

Mark Blyth

November 25, 2020

1 How does standard CBC discretisation work?

1.1 Problem formulation

This follows sections 4 and 4.1 of [1]. Assume our periodic orbits form the solution family to some ODE

$$\dot{y} = g(y, \mu) , \quad (1)$$

for state $y \in \mathbb{R}^n$, parameter $\mu \in \mathbb{R}$. Let $\Gamma = \{(y_s, \mu_s, T_s) : s \in \mathbb{R}\}$ be a family of 1-periodic orbits y_s , with parameter value μ_s and index-variable s . Entries $(y_s, \mu_s, T_s) \in \Gamma$ are defined such that the T_s -periodic function $y_s(\cdot/T_s)$ is a solution to equation 1 at parameter value μ_s .

CBC aims to trace out the family Γ of periodic orbits. That is, given some initial $(y_0, \mu_0, T_0) \in \Gamma$, we wish to continue the family and find other orbits $(y_i, \mu_i, T_i) \in \Gamma$. Let $U(\Gamma)$ be a neighbourhood of Γ , containing both the solution family, and near-by periodic orbits that are *not* a solution to 1. Assume now that our system state y is controlled using a proportional control strategy. Given some T_i -periodic control target $\tilde{y}_i(\cdot/T_i)$ and parameter μ_i , for $(\tilde{y}_i, \mu_i, T_i) \in U(\Gamma)$, the output of the controlled system will be some function $y_{\text{out}}(\cdot/T_i)$. The controlled system therefore acts as a map. Lazily denote this map by $Y : U(\Gamma) \rightarrow y_{\text{out}}$. Note that, as the output is some measured observation of the system, and the control target specifies our desired value for this output, $y_{\text{out}} \in \mathbb{R}^{m \leq n}$ and $\tilde{y}_i \in \mathbb{R}^m$. Y is referred to as the IO map, and dictates the relationship between control target input and measured system output. The IO map is evaluated by setting the system parameter to μ_i and the control target to $\tilde{y}_i(\cdot/T_i)$; running the controlled system until transients have died away; then measuring the output of the system.

A control target y_i^* is noninvasive if and only if y_i^* is a fixed-point of the IO map, satisfying $Y(y_i^*, \mu_i, T_i) = y_i^*$. In this case, the actual system

output exactly matches the target system output. When using a proportional control strategy, zero tracking error means zero control action, which in turn means the system is acting under its own free dynamics, and therefore control is noninvasive. CBC aims to find the family of targets y^* that produce a fixed-point of IO map Y when using a proportional control strategy. This cannot be achieved directly; instead, a discretised problem must be solved.

1.2 Discretisation method

This follows section 4.2 of [1].

At any given continuation step s , we wish to solve $Y(y_s^*, \mu_s, T_s) = y_s^*$ for the noninvasive control target y_s^* , where $(y_s^*, \mu_s, T_s) \in U(\Gamma)$. We seek the fixed-point of map Y , however this is not a numerically tractable problem. Instead, candidate solutions y are discretised using a Galerkin projection Φ . Given some candidate solution $y(\cdot/T_s) \in \mathbb{R}^m$, we project onto the first q Fourier modes, such that we have a discretisation projector $\Phi : C_1 \rightarrow \mathbb{R}^{(2q+1)m}$, for C_1 the set of 1-periodic functions. We now substitute our control target with its discretised approximation $\sum \phi_j b_j(t)$, for Fourier basis functions b_j , and pass this reconstructed control target through Y . Finally, we again apply the projection Φ to the output of the IO map, yielding a discretised map $\phi^{\text{out}} = \Phi \left[Y \left(\sum \phi_j^{\text{in}} b_j(t), \mu, T \right) \right]$ mapping input coefficients ϕ^{in} to output coefficients ϕ^{out} . We then seek the discrete fixed-point $\phi^{\text{in}} = \phi^{\text{out}}$.

2 Where does CBC discretisation break down?

Instead of discretising the map, we discretise the input and output signals, and seek the discretised input that, when undiscretised, leads to noninvasive control. Evidently, the correct noninvasive solution y^* can only be found when $y^* \in \text{span}\{b_1, b_2, \dots, b_n\}$. We can only find the true solution when we can express it using our choice of basis functions. We must therefore either select basis functions that span the set of valid solutions to the fixed-point problem, or otherwise prove that the discretisation method will approximate the solution to a high degree of accuracy. A sufficiently high-dimensional Fourier discretisation will allow us to represent any periodic signal, and hence any valid solution, to an arbitrary degree of accuracy. Nevertheless, this guarantee may not hold when choosing other basis functions. For example, an order-3 BSpline model is only smooth up to its second derivative. An ODE with a C^k -smooth right-hand side will have a C^{k+1} -smooth solution*.

A third-order BSpline solution will therefore be inaccurate for any finite number of basis functions, when the system is governed by a $C^{k>1}$ -smooth ODE. Toy neuron models are C^∞ -smooth, meaning one cannot find an exact IO map solution using finite numbers of BSplines, and we must instead hope that a sufficiently accurate approximate solution exists. A counter-argument would be that in general, a Fourier series only converges to the target signal in the infinite basis function limit, yet we are happy to accept a truncated Fourier series as being accurate to within working precision. Perhaps the rate of error decay is more important, in which case other basis functions may indeed be preferable to Fourier.

ZZZ * I think this is true

Let us now consider the validity of the method when discretisations are inexact. Say the true noninvasive solution does not lie within the span of our basis functions. Consider the following thought experiment.

- Take a physical system.
- Run it, without a controller, until it converges to a natural periodic orbit.
- Measure a system output of interest.
- Discretise that system output.
- Use this discretisation to create a control target.
- Control the system using this target.
- Again, run to convergence and measure the output.

This is very similar to how one would initialise a control-based continuation, and as such, represents a realistic scenario. As we are trying to control the system towards one of its natural periodic orbits, we should straight away have noninvasive control, simply by taking the measured system output as our control target. However, since the system output does not lie within our basis function subspace, the discretisation will be inexact. That is, we do not return to our original signal when discretising then undiscretising, but instead we reach some approximation of it, with this approximation given by the projection of the signal onto the basis functions. In this case, while our natural system response is noninvasive and hence a solution to the continuous problem, the signal discretisation is *not* a solution to the discretised problem.

This occurs because, as a result of the inexactness of the discretisation, there will be a difference between the natural periodic orbit and the control target obtained through its discretisation. Hence, the controller will actually push the system away from the natural orbit, and towards the control target. For the controller to push the system away from the orbit, it must have a non-zero control action. This non-zero control action results in a proportional error – a difference between the control target and the actual system output. As the input and output signals are not equal, their discretisations are not equal, and so the input is not a fixed-point of either the continuous or the discretised IO map, even though the input was obtained from a natural system response, *and therefore a solution to the continuous IO map*.

In summary, we started from a natural periodic orbit of the system. Such an orbit is a solution to the continuous fixed-point problem, and represents a noninvasive control target. Our discretisation is inexact; as a result, the reconstructed continuous-problem solution, as obtained from its discretisation, does not equal the actual continuous-problem solution, and neither the discretisation nor its corresponding function are a solution to their respective fixed-point problems. As soon as our discretisation has any error, a discretisation of a solution to the ‘full’ continuous problem is *not* a solution to the discretised problem. This makes it impossible to exactly solve the continuous problem through solving a discretised problem, when there is any error to our discretisation. A key question is whether this issue is significant or not: one may hope that the solution to the discretised problem is still noninvasive to within working precision, however it is not clear to me how we could prove or disprove this.

ZZZ This method of discretisation is a Galerkin projection. It is a common method in PDEs. I’m currently digging into the literature on it, to make sure I fully understand everything about the method before making any strong claims about its performance.

3 How would CBC collocation work?

3.1 Collocation for boundary value problems

See here for a short intro to collocation methods for boundary value problems. While not used in this work, [2] has a thorough description of collocation for the periodic boundary value problems typically encountered in continuation. The fundamental idea is, of course, the same as that discussed here.

Collocation allows one to obtain a function that approximately solves

a boundary-value problem. It is a finite-elements method; an alternative approach would be finite-differences, which yields a set of points that lie on the solution curve, without producing a model of the curve itself. Finite element methods select a set of basis functions B_i , who span the subspace in which we wish to approximate the solution.

For simplicity, we here consider the one-dimensional, linear case. Let L be some linear differential operator, and $f(t)$ be the inhomogenous component of our boundary value problem. We then have

$$\begin{aligned} Ly(t) &= f(t), \quad t \in [a, b], \\ y(a) &= \alpha, \quad y(b) = \beta. \end{aligned}$$

We choose a solution approximation with form $y(t) = \sum_{i=1}^n \beta_i B_i(t)$, for unknown coefficients β_i . Applying operator L to this gives $Ly = \sum \beta_i LB_i$, and our approximation of the operator problem becomes

$$\begin{aligned} Ly &= \sum_{i=1}^n \beta_i LB_i, \quad t \in [a, b], \\ \sum_{i=1}^n \beta_i B_i(a) &= \alpha, \quad \sum_{i=1}^n \beta_i B_i(b) = \beta. \end{aligned}$$

We now want to find the unknown coefficients β_i that solve this approximate problem. To do this, we partition the time-domain into $n - 2$ collocation points, and seek coefficients such that our approximate solution exactly solves the operator problem at these points. Let $\Xi = [a < \xi_1 < \xi_2 < \dots < \xi_{n-2} < b]$ be our collocation mesh; the collocation system then becomes

$$\begin{aligned} \sum_{i=1}^n \beta_i B_i(a) &= \alpha, \\ \sum_{i=1}^n \beta_i LB_i(\xi_j) &= f(\xi_j), \quad j = 1, 2, \dots, n - 2, \\ \sum_{i=1}^n \beta_i LB_i(b) &= \beta. \end{aligned}$$

This is a set of n equations, 2 boundary conditions and $n - 2$ collocation conditions, for the n unknown coefficients β_i . Owing to the linearity assumption, this can be solved using simple linear algebra. In general, we will not have a linear operator L , in which case the collocation system must

instead be solved using some nonlinear solver. Coefficients β_i discretise the solution, and can be used in conjunction with their respective basis functions to evaluate the approximate solution at any point across its domain.

3.2 Collocation for CBC

Consider the binded IO map $Y(\cdot, \mu_s, T_s)$, bound to some predetermined parameter value μ_s and signal period T_s . The map acts as a nonlinear operator, acting on control target y_{in} to produce some signal y_{out} . Let us denote this operation by $y_{\text{out}} = N(y_{\text{in}})$. We wish to solve for the fixed-point $y^*(t/T_s)$ of this map, thus giving a nonlinear operator problem on y^* . As before, select a set of n basis functions B_i , such that our approximate solution is given by $y^*(t/T_s) = \sum_{i=1}^n \beta_i B_i(t/T_s)$. Note that y^* is, by definition, a 1-periodic function. To construct our collocation problem, we therefore divide the time-domain into a collocation mesh Ξ , where

$$\Xi = [\xi_1 = 0 < \xi_2 < \dots < \xi_{n-1} = 1] .$$

Enforcing periodicity, our collocation equations become

$$N \left[\sum_{i=1}^n \beta_i B_i(t/T_s) \right] = \sum_{i=1}^n \beta_i B_i(t/T_s) , \quad t = \xi_1 T_s, \dots, \xi_{n-1} T_s$$

$$\sum_{i=1}^n \beta_i B_i(0) = \sum_{i=1}^n \beta_i B_i(1) .$$

Here, we have reconstructed our approximate solution from the set of basis functions; rescaled to a T_s -periodic function; passed this through the nonlinear operator N ; and mandated equality between the input and output functions at a set of period-independent mesh points. Period-independent refers to the fact that each mesh point will always align to the same point along the signal phase, independently of the period of the signal. We have also enforced continuity between periods. This system gives n equations for the n unknowns β_i , and can be solved using a nonlinear solver. The resulting β_i produce a signal discretisation that can be embedded within a pseudo-arclength predictor-corrector scheme; they can also be used, in conjunction with their respective basis functions B_i , to produce a continuous model of the solution, which can be used as a control target.

3.3 Collocation modifications

For autonomous systems, any phase-shifted solution is also a solution. This results in an indeterminacy within the collocation system. We must therefore impose a phase condition, to enforce uniqueness on the solution. The most reliable phase condition [2] is the integral phase condition

$$\Psi[y^*] = \int_0^1 \langle y^*(t), \dot{u}(t) \rangle dt = 0 ,$$

for reference solution $u(t)$. One would typically choose the periodic orbit obtained at the last continuation step for the reference solution. The integral phase condition selects the periodic solution whose phase aligns with that of the reference solution, by selecting the phase shift that minimises the distance between the reference and current solutions. As we have added an additional constraint, we must either remove one of the collocation meshpoints, or include an additional basis function, to ensure a well-posed problem.

If our basis functions are 1-periodic, the periodicity constraint is automatically satisfied, and thus provides no additional information. We then have one more unknown than we have equations. For our system to be well-posed, we must either add a collocation point, to increase the number of constraints by one, or remove a basis function, to reduce our number of unknowns by one. The periodicity constraint can be removed in such cases.

4 Differences between Galerkin and Collocation CBC

When using standard Galerkin discretisation, we seek the basis function coefficients that remain unchanged when their corresponding signal passes through the IO map. For collocation discretisation, we instead seek coefficients such that the input and output signals remain unchanged at the mesh points. As such, the basis function coefficients are the object of interest in Galerkin discretisation, whereas they are simply a means to an end with collocation. Similarly, the input and output functions are a means to an end with standard discretisation, as we are not interested in the functions themselves values, so much as the basis function coefficients that produce those functions. The opposite is true of collocation: the input and output functions are the object of interest. We need only consider the basis function coefficients insofar as to produce a valid input function.

Standard CBC requires us to project the output signal onto the basis functions, for example using FFT. We never perform any projections when using collocation, as we only ever need to transform from coefficients to

signal. This will offer a minor computational speedup to the collocation method.

If the discretisation is inexact – if there exists a difference between the true function and its discretised approximation – we cannot necessarily find a solution using Galerkin discretisation. Furthermore, equality between input and output coefficients does not guarantee noninvasive control. With collocation, we can always find a solution – even when the discretisation is inexact – however this solution again does not guarantee noninvasive control.

5 Where might CBC collocation break down?

Collocation solves for equality between the input and output functions, at a set of chosen phase points. If the output function is noise-corrupted due to measurement errors, equality would be difficult, if not impossible, to find. Various approaches could be used to overcome this. Firstly, the output could simply be replaced by a surrogate. If the surrogate is accurate, the issue is solved. Alternatively, one could approach the problem statistically, and solve for the maximum-likelihood basis function coefficients. By explicitly modelling noise, the solution should be more robust against experimental error. Finally, one could average the samples across many periods, and attempt to remove experimental noise in this way.

Another issue would be that of overfitting. We are only solving the fixed-point problem at a selection of collocation meshpoints. The assumption is that between these meshpoints, the resulting solution is still a good approximation of the true solution. Nevertheless, there is the risk that the collocation solution would deviate strongly from the true solution between meshpoints. In this case, the collocation solution would not be suitable for use as a control target. Surrogates may once again be able to assist here, by smoothly interpolating between the accepted collocation solutions points. Alternatively, choosing more collocation points would reduce the maximum error between each point, however this comes at the expense of increasing the discretisation dimension.

6 End note

I'm currently fitting the BSpline basis function coefficients using least-squares on the sampled system output. If we had a continuous function, we could project onto the basis functions using an integral-based inner product. This could be approximated using a numerical quadrature method. It would be

interesting to investigate the similarities and differences, if any, between samples-based least-squares and functions-based inner-product projection.

References

- [1] Jan Sieber and Bernd Krauskopf. Control based bifurcation analysis for experiments. *Nonlinear Dynamics*, 51(3):365–377, 2008.
- [2] Yuri A Kuznetsov. *Elements of applied bifurcation theory*, volume 112. Springer Science & Business Media, 2013.