Discretisation: the beginning of the end?

Mark Blyth

Summary

Fourier/Galerkin discretisation is inefficient for neuronal signals, so we need something better

Last time:

- BSpline/Galerkin is numerically finicky
- Orthogonal collocation cold be a suitable alternative method

This time:

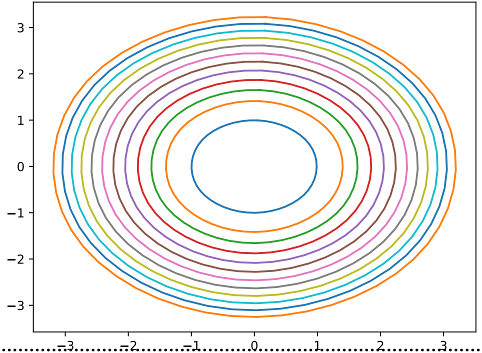
- Collocation progress
- Progress on BSpline finickiness
- Work plan for the year

Collocation progress

- Implemented ‘standard’ orthogonal collocation
  - Lagrange polynomial basis functions, no control aspect
- Reformulated for BSpline basis functions
  - Using Lagrange polynomials gives a picewise-polynomial solution, much like splines
  - Key difference is spline basis includes smoothness requirements too, potentially useful for CBC
  - Spline knots define solution mesh, simplifying problem slightly
  - Numerically, both work nicely; can’t rigorously say which one is better or worse
- Finished during Christmas; yet to build into a control-based continuation

Collocation progress

Numerical continuation of periodic orbits from a Hopf normal form



Testing BSpline/Galerkin

BSpline/Galerkin struggles on some parts of the solution curve; why? Tested...

- Control gains: no major impact, within a sweetspot
  - Big enough to stabilise UPOs, small enough to preserve numerical accuracy
- Solvers: SciPy and my DIY Newton
  - No difference, good to see; my solver is significantly slower
  - Suggests issues are from prediction/correction setup or existence-and-uniqueness, rather than Jacobian estimation
- Stepsizes: has a big impact with Fourier/Galerkin
  - Fourier/Galerkin can be made to fail in the same way as BSpline/Galerkin, when bad stepsizes are chosen
  - BSpline/Galerkin can't be made to work well when varying stepsizes
  - *Maybe an adaptive stepsize is needed!*

Adaptive stepsize methods

- Consider a prediction $p(h)$, obtained for stepsize h
- Newton correction $c(h) = p(h) - J_f^{-1}|_{p(h)} f\left(p(h)\right)$
- Size of first step $\delta(h) = \|J_f^{-1}|_{p(h)} f(p(h))\|$ estimates the error of the prediction
- If δ is too big, shorten the stepsize and try again
- If δ is too small, lengthen the stepsize for next time
- Bonus: use an asymptotic expansion to choose the best stepsize

Adaptive stepsize methods

- We can quantify the 'speed of approach' with contraction rate κ

$$\kappa(h) := \frac{\|\text{second Newton step}\|}{\|\text{first Newton step}\|}$$

- By asymptotic expansion, $\kappa(h) = \varkappa h^2 + \mathcal{O}(h^3)$

Strategy:

- Choose target contraction rate
- After each step, estimate the stepsize $h = \sqrt{\frac{\kappa}{\varkappa}}$ that would have given our contraction rate
- Use that, stepsize from δ asymptotic expansion, and current stepsize to choose next stepsize

Adaptive stepsize results

- Monitoring contraction rate, and size of the first Newton correction
- This gives a lot of extra hyperparameters: min, max stepsize; initial stepsize; nominal contraction rate and predictor error
  - Getting results means choosing sensible values for all of these, which isn't easy!
- Can't use the adaptive method with a pre-rolled solver; Newton Jacobian estimation is painfully slow; takes a long time to test hyperparameters
  - Broyden update is the way to go

No results so far; Newton solver diverges at first fold bifurcation, with or without adaptive stepsizes

Solver divergence

Issue: Newton solver diverges at first fold; doesn't happen with Fourier discretisation

- Wasn't previously an issue as I'd run only 1 Newton step
  - Interesting that it does happen; probably a result of the control aspect
- Convergence criteria: $\|x_n - x_{n-1}\| < \mathrm{tol} \;\; or\, f(x_n) < \mathrm{tol}$
  - With Fourier, $x_n, f(x_n) \in \mathcal{O}(\mathrm{small})$
  - With BSplines, $x_n, f(x_n) \in \mathcal{O}(1)$
- Using the same tolerance implicitly applies looser convergence requirements to Fourier
- Proposals:
  - Use relative tolerances instead, eg. $\sum_i \left(\frac{x_n^i - x_{n-1}^i}{x_n^i}\right)^2$
  - Or, take the best solution from the first n iterations

Immediate plan

Demonstrate how BSplines can be used for efficient CBC on slow-fast systems

- Check that adaptive stepsizes *do* make splines work
- Switch Duffing for van der Pol oscillator
- Implement an appropriate (CBC-inspired or numerical-inspired) phase constraint
- Implement intelligent / adaptive BSpline knot selection
  - BSpline knots generally need careful placement to be an efficient discretisor
- If it all works, write it up!

Perhaps focus more on how CBC can be used on slow/fast systems, and less on discretisation

Mid-term plan

Lots of other discretisations could work

- Try collocation, wavelets, surrogate-based
- Produce a recipe book of discretisation methods, suggesting which to use when
- Develop an algo for the experiment to choose its best discretisor at each step?

Covers similar research to the other proposed paper, challenge would be making it a unique contribution

Long-term plan

Automated neuronal identification and classification

- Option 1: classify bursters from their fast subsystem bifurcations
  - Approach 1: try to implement slow/fast analysis methods in a CBC framework
  - Approach 2: use feedback control to gather data for fitting cubic Lienard model; analyse fitted model to extract classification
  - Challenge: can't study each subsystem individually, on a real experiment
- Option 2: couple CBC to model identification procedure, and fit a 'generic' HH-model
  - Can hopefully discover a cell's ion channels and their kinetics, without any a priori knowledge
  - Challenge: lots of different gating and conductance dynamics; a general model might be too general to accurately fit
  - Simplification: use voltage, current, dynamic clamp results as prior information; CBC then becomes an enhanced model fitting method

Some questions

- Are these ideas biologically useful?
  - Burster classifications are interesting mathematically, but are they of biological significance?
  - Is a classification experiment of interest to experimenters, or is it more a mathematical toy?
- Lots of interesting dynamics can appear in bursters and multi-timescale systems
  - Mixed-mode oscillations, canards, torus canards, noise-induced bursting
  - Are these dynamics important biologically, or are they more mathematical curiosities?
  - Would slow-fast CBC be missing key biological dynamics by ignoring these behaviours?
- Is slow/fast enough? Do we need additional (medium, or super-slow) timescales?
  - Seen some papers using 3 timescales; are two-timescale models too simple to capture real dynamics?
- Are burster classifications limited to single cells, or could the same methods reveal information about networks?

Next steps

This wek: NODYCON slides and presentation; then…

- Try adaptive stepsizes to demonstrate splines success on Duffing oscillator
- Generalise code to work on van der Pol oscillator
  - Implement a phase constraint, and knot selection
- Test it all out!