Continuation and polynomials

Mark Blyth

Week’s goal

Code up a numerical continuation algo

- Goal: use BSplines for the continuation
- Result: skimmed a pile of papers
  - Dropped down a rabbit hole, but a more relevant one than usual!

Continuation coding

Following the continuation algo described in Kuznetsov Elements

- Uses Lagrange polynomials as continuation basis functions
- It states we should choose zeros of Legendre polynomials as collocation mesh
  - Provides maximal accuracy at collocation points

Lagrange polynomials

Not provided by SciPy in the required form, so I need to implement them myself

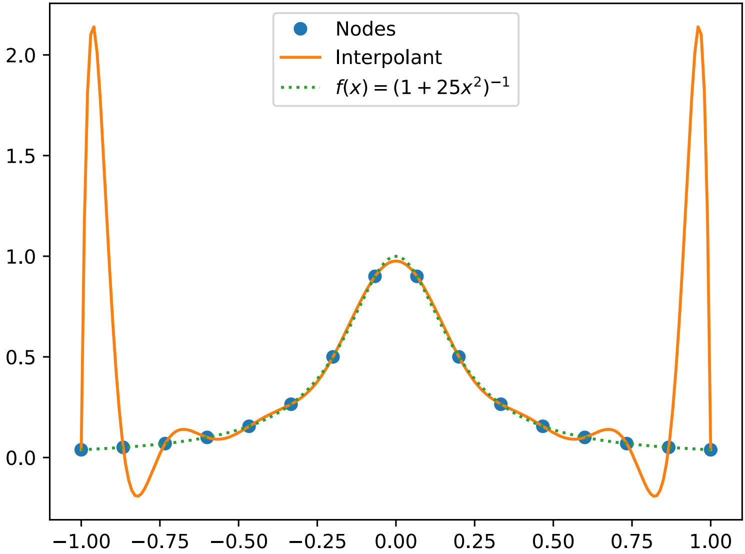
- $f(t) = \sum \beta_j l_j(t)$
- $l_j(t) = \prod_{m \neq j} \frac{t - t_m}{t_j - t_m}$
- Inefficient: $\mathcal{O}(n^2)$ flops for each t evaluation

Barycentric Lagrange polynomials:

- Denominator is t-independent, so pre-compute it as weights w_i
- Compute t-dependent product $\omega(t)$ at each evaluation
- $f(t) = \omega(t) \sum \beta_i w_i \frac{1}{t - t_j}$
- $w_i = \frac{1}{\prod_{k \neq i}(t_i - t_k)}$
- $\omega(t) = \prod(t - t_k)$
- $\mathcal{O}(n)$ flops for each t evaluation

Lagrange polynomials and CBC collocation

Lagrange polynomials are heavily susceptible to Runge's phenomenon



Runge’s phenomenon and Lagrange polynomials

- Standard setup:
  - Use Lagrange interpolating polynomials
  - Use zeros of Legendre polynomials as collocation mesh
- Claimed to give best accuracy *at* collocation points, but what about between them?
  - If we’re using the result as a control target, Runge’s phenomenon is not acceptable
  - We need accuracy *between* collocation points, just as much as *at* meshpoints

Runge's phenomenon

Idea: use Chebyshev nodes as collocation points

- Minimises $\|f(t) - p(t)\|_\infty$, the largest deviation between a continuous function f and its polynomial approximation p
- Minimises Runge's phenomenon!
- Could use Chebyshev nodes for Lagrange collocation in CBC
- Equivalently, could use Chebyshev polynomials as collocation basis functions
  - Chebyshev polynomial collocation exists!
  - The paper on Chebyshev collocation looks very useful; yet to read it
- Will *hopefully* make the collocated solution a good control target

Two notes

- Splines let us control the order of the polynomials by splitting the function up into separate polynomial segments; splines therefore also control Runge's phenomenon

- Lagrange, Chebyshev polynomials form an orthonormal basis
  - Could use them in place of Fourier or BSplines for Galerkin CBC
  - Based on Runge's phenomenon, they might not be a good choice for neuronal signals
  - Could work very for 'simpler' (Duffing) systems

Main take-aways so far

- Currently coding up standard numerical continution with Lagrange polynomial collocation, as per Kuznetsov Elements
- For CBC applications, we don't want Runge's phenomenon
  - Chebyshev might be better than Lagrange polynomials for CBC
  - Splines might be better than interpolating polynomials
- Could use Lagrange, Chebyshev polynomials for either 'standard' Galerkin CBC discretisation, or CBC collocation

NODYCON reviewer 2

Their suggestion: fit a ‘proper’ model of the system, and use that as a surrogate

- Issue: requires us to come up with some generic model that we can fit to the system; hard to do if we don’t yet know what the system does
- Refinement: combine system identification and surrogate modelling
  - Simultaneously produce and refine a model of the system, and use that as a surrogate for further continuation steps

- Brought to mind reproducing kernel Hilbert spaces
  - Kernel method, like GPR: projects into a feature space; models are linear in feature space, nonlinear in original space
  - Used in ML for fitting regression models; would work as a surrogate
  - Used in NLD for system modelling and identification
  - If it’s both a regression model and a system identification method, maybe it’s exactly what we need?

I don’t yet understand anything about RKHS. Going to work through some papers and figure out if they’ll be useful or relevant.

Next steps

- Keep coding up ‘standard’ numerical continuation
- Try Legendre, Chebyshev, BSpline, . . . CBC collocation with both standard continuation and CBC
  - Compare solution curves for collocation, Galerkin, and various basis functions
- Try Galerkin CBC again
- See if RKHS do anything interesting

References

- Bouvrie, Jake, and Boumediene Hamzi. "Kernel methods for the approximation of nonlinear systems." SIAM Journal on Control and Optimization 55.4 (2017): 2460-2492.
- Nejib, Hamza, Okba Taouali, and Nasreddine Bouguila. "Identification of nonlinear systems with kernel methods." 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). IEEE, 2016.
- Hamzi, Boumediene, and Houman Owhadi. "Learning dynamical systems from data: a simple cross-validation perspective." arXiv preprint arXiv:2007.05074 (2020).
- Berrut, jean-paul, and lloyd n. trefethen. "barycentric lagrange interpolation." siam review 46.3 (2004): 501-517.
- Nosedal-Sanchez, Alvaro, et al. "Reproducing kernel Hilbert spaces for penalized regression: A tutorial." The American Statistician 66.1 (2012): 50-60.
- Daumé III, Hal. "From zero to reproducing kernel hilbert spaces in twelve pages or less." Online: `http://pub`. hal3. name/daume04rkhs. ps (2004).
- Wright, Kenneth. "Chebyshev collocation methods for ordinary differential equations." The Computer Journal 6.4 (1964): 358-365.