University of BRISTOL

[2021-01-25 Mon]

More BSpline struggles

Mark Blyth

Week's work

Last time:

- Adaptive stepsizes might fix everything

This time:

- Adaptive stepsizes are lots of hassle for little benefit
  - BUt that's an interesting insight in itself!
- Jacobian computation has a large impact on results

Some convergence issues

- Before adaptive stepsizes, I used a single Newton iteration for speed
  - Adaptive stepsizes requires 2+ Newton iterations; ran until convergence, instead of taking a single step
- With more Newton steps, the iterations diverge at the fold
  - Taking more steps leads to exponentially more wrong solutions
  - Also implemented Newton-Broyden; same thing happens
- The same thing happens with and without adaptive stepsizes
  - Taking a single Newton step works
  - Taking more steps causes exponentially fast divergence at the fold
- Stepsizes adapt a lot, suggesting convergence properties change rapidly along the curve

Hypothesis 1

BSplines are a bad way of representing the signal

- Fourier is a natural description of the signal; fundamental harmonic describes amplitude, the rest describe shape
- Maybe splines don't caputure the signal well, and
- If so...
  - Small changes in the signal give big changes to BSpline discretisation
  - Continuation curve is very wiggly in discretisation-space?
  - Tangent prediction becomes a fairly useless starting point

Test: look successive BSpline coefficients

Result: they change nice and smoothly; probably not an issue; hypothesis rejected

Hypothesis 2

The system has not converged to a stabilised PO at divergence points

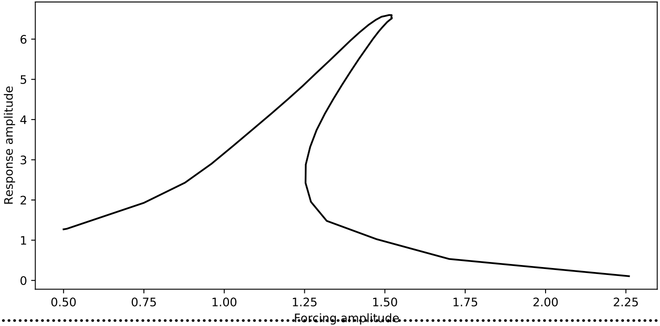
- Doesn't make sense to discretise transients
- If the system hasn't converged, the IO-map evaluation, and therefore predictor-corrector calculations, are meaningless

Test: plot the control target and system output after tangent prediction

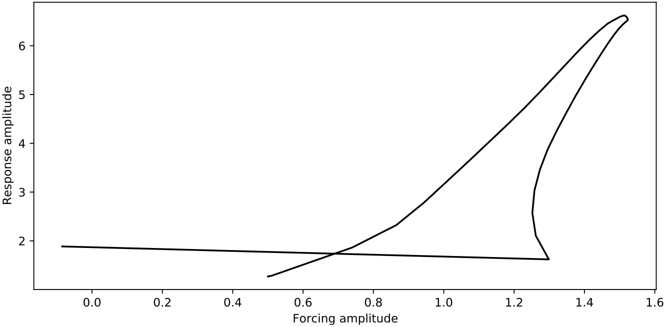
Result: at tangent-prediction, the PO has always converged; probably not an issue; hypothesis rejected

Other problem

Struggles to converge on second SPO branch; jumps randomly, happens to be in the right direction

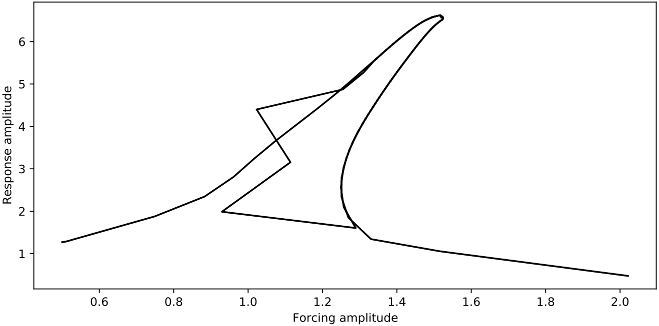


Faster Jacobian method



Identical setup but slightly different Jacobian computation

Even faster Jacobian method



Another identical setup, new Jacobian. Looks like solution repels the Newton interations; doesn't entirely make sense

Hypotheses 3, 4

Solution fails to converge, and jumps randomly

- Sometimes we get lucky and it jumps back to the curve
- Not really working properly!

Either

- Jacobian is somehow problematic
  - First Newton step succeeds, so initially the Jacobian is probably right

or

- Continuation equations are misbehaving
  - Broyden only uses the initial Jacobian, and updates from function values
  - Broyden shows same divergence; presumably it's the function values at fault

Computational setup

All approaches use single Newton iterations; difference here is in the Jacobian computation; three methods tested:

- Pre-made numdifftools
  - Adaptive FDSS; should give best results
  - Slow; 1h 8 minute runtime
- Pre-made numdifftools
  - Fixed FDSS; more potential for inaccuracy
  - Faster run-time
- Simple DIY finite differences with fixed FDSS
  - Forward or central-step finite differences
  - 7 minute run-time

Jacobian computation

Forward:

$$J[i,j] = \frac{f_i(x + he_j) - f_i(x)}{h}$$

Central:

$$J[i,j] = \frac{f_i(x + he_j) - f_i(x - he_j)}{2h}$$

Forward: $n+1$, central: $2n$ function evaluations, for $x \in \mathbb{R}^n$

Jacobian accuracy

Changing FDSS has a big effect on the Jacobian

- Needs to be very large to get results reliably
- Typically would use $\mathcal{O}(10^{-6})$ steps; I use 0.2
- Changing stepsize has a large impact on Jacobian

Changing between central and forward has a big effect on Jacobian

- Changing between forward and central changes some entries by 10%

Can't reliably take correct Newton steps if we can't find an accurate Jacobian

Some issues

- No idea why this happens with splines but not Fourier
- Can't spot an easy way to test if the Jacobian is the problem
  - Broyden results suggest its more likely down to the continuation equations
  - Misbehaving continuation equations would also make it harder to compute a Jacobian
- If Jacobian isn't the problem, the continuation equations are misbehaving
  - Eg. solution has a very small basin of attraction
  - This is easier to test: try collocation – different continuation equations
- Different continuation equations might help both the solution behaved-ness and Jacobian computation

Next steps

This week:

- Reading, writing, NODYCON presentation

Then…

- Ignore the problem!
  - Tried lots of ideas and it's still not working properly
  - I'm not convinced I can gain any further insights with the current simulate-and-test method
- Implement a phase condition, and test BSpline CBC with a different system
- If that doesn't work either, try BSpline collocation
  - All the usual BSpline benefits
  - Hopefully more numerical stability
  - Less noise robustness (but that can be overcome!)