Tags: #logbook - Denison
Links: Logbook_01_220115
Files: `01_substitution_220117.nb`, `PQSHelper.wl`

Logbook_220117

A Numeric Project

> **Aims**
>
> - [ ] Experiment with different frequency shifts
> - [ ] Experiment with different positions in the cycle
> - [ ] See how much the output changes (since only some of the light leaves at the output coupler)

A.1 Notes

- In reality, the frequency shift applied by the acousto-optic effect would be around 10 MHz
  - This is much less than the frequency resolution of our data (which is 2.5 GHz)
  - So therefore the Fourier transform method used in Logbook_01_220115 isn't really realistic
- Instead, just multiply E by $\exp(-2\pi i f t)$

A.1.1 Output Frequency

Even though we may be applying a frequency shift of ~ 1 GHz (for example) each round trip, we won't get a 1 GHz shift every trip for the output pulse of the laser

- Because (based on the output coupling, which is default at 0.5), a certain number of photons leave the laser at each round trip

    - And so if `feedback = 0.5`, we expect that on average a photon will undergo two round trips before exiting
    - Therefore only a shift of 2 GHz
- This should be encapsulated by the `gain_PQS.m` function
    - The gain profile used is centred on the original wavelength, not the peak wavelength

A.2 Results

A.2.1 Improved Frequency Shift

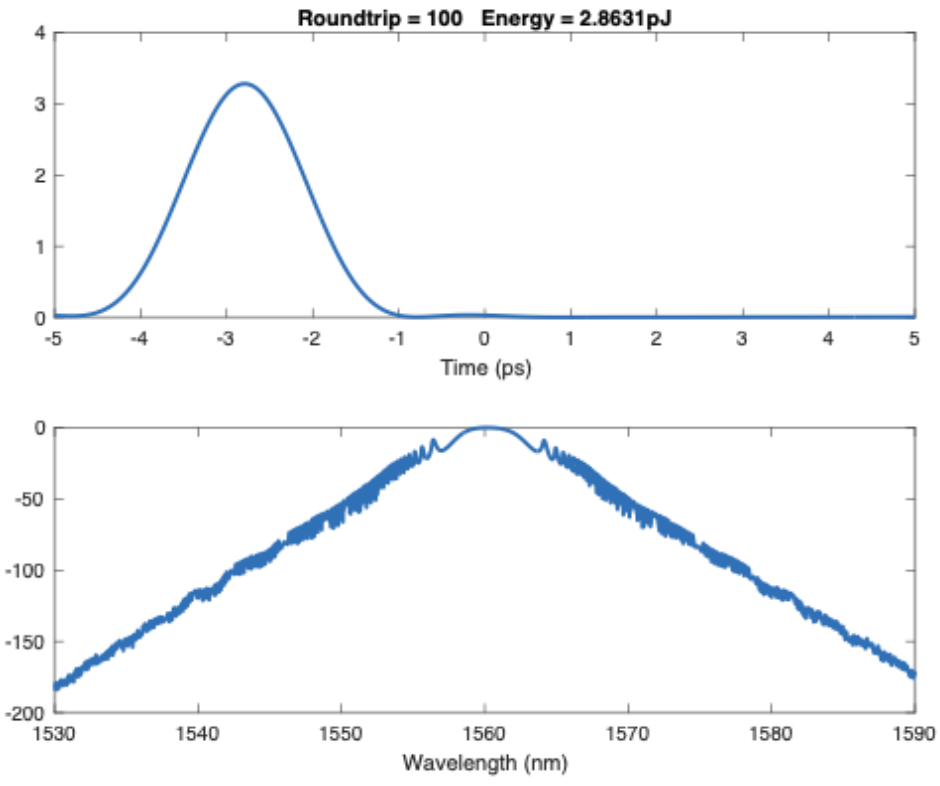
Changed `FreqShift.m` to use the multiplication method:

```
E = E .* exp(-1i * freqdelta * 2 * pi * t);
```

- This should be faster, and also allow for any continuous frequency instead of just multiples of the resolution.

Observed: (using `freqdelta = df`)

`01__202201181612.fig`



- Produced the same result as `202201171130.fig` using the FFT method
  - As expected
  - (Note that the line thickness is increased here, but the data is the same)

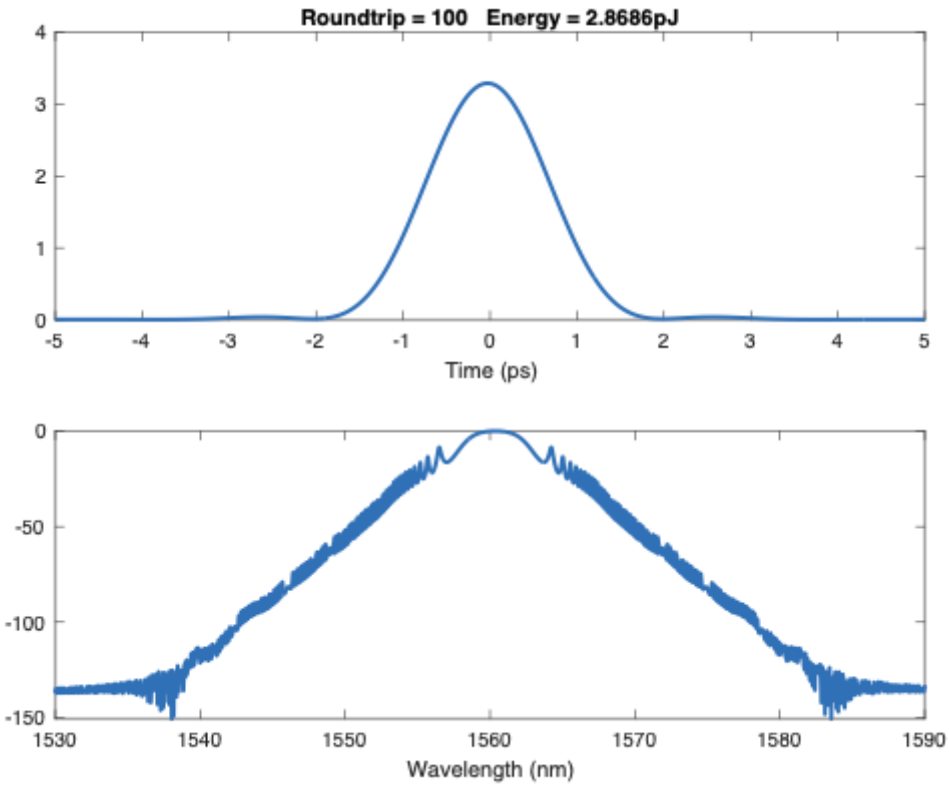
A.2.2 Different Frequencies

How does the results change with different frequency shifts?

- We saw last week that too large of a shift results in no soliton forming
  - But it could actually just be that the soliton moves off the limits of the time graph

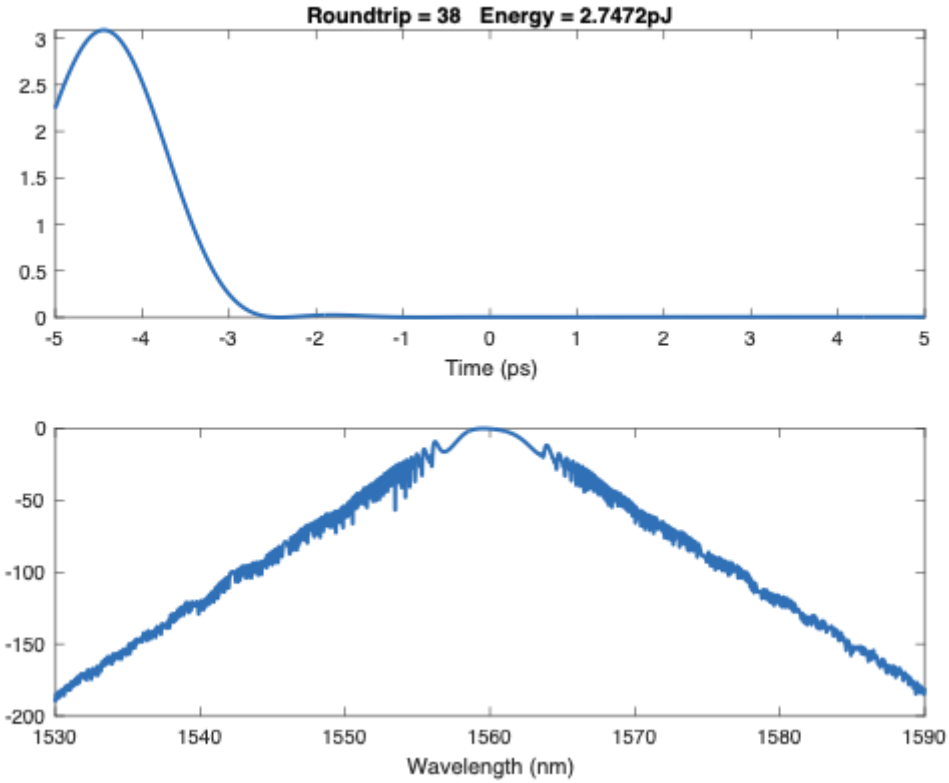
Using `freqShift = 1e7`, we get

```
01__202201201040.fig
```



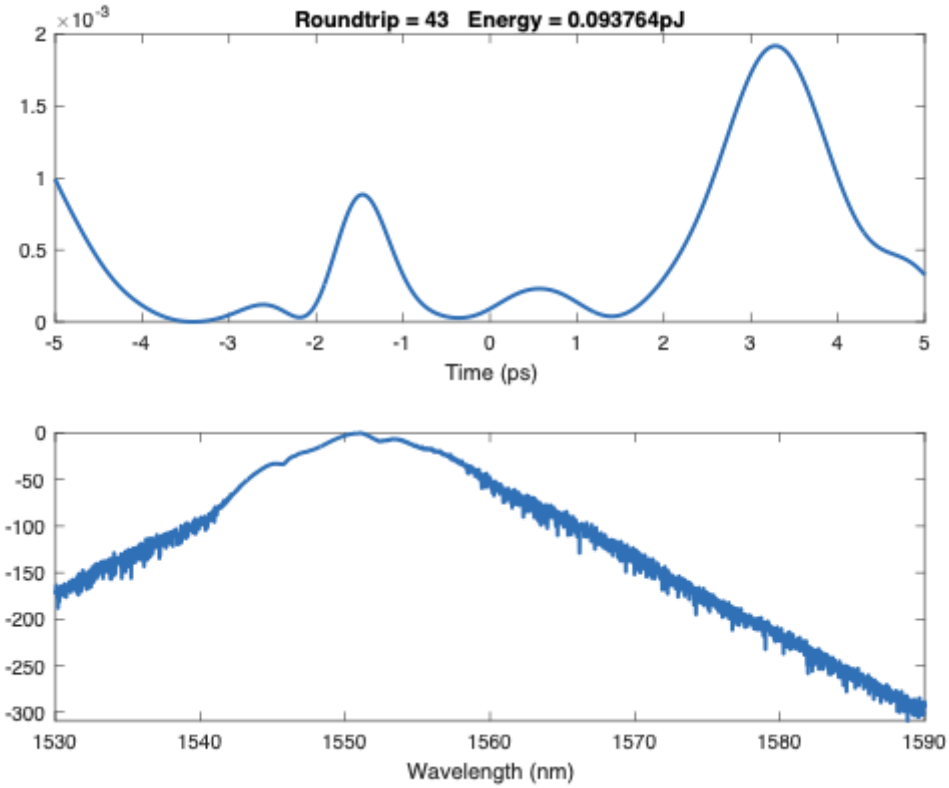
- Qualitatively the same as the unmodified soliton

Using `freqShift = 1e10` we get

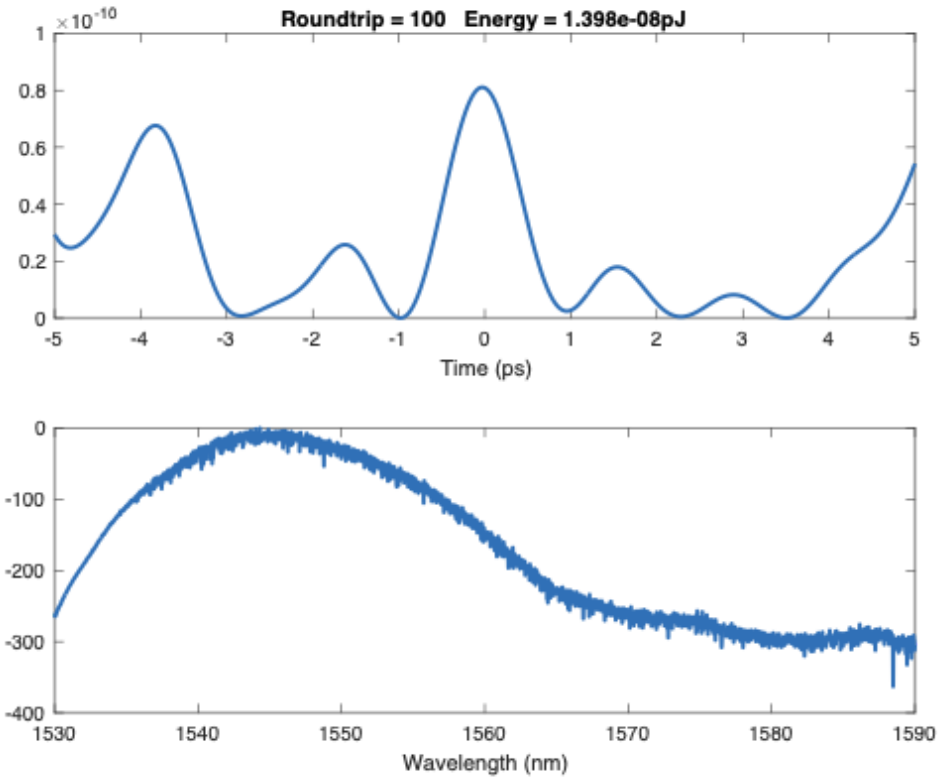


- Took the capture early, so that the pulse doesn't move off the axes

Using `freqShift = 5e10`, we get



`01__202201201050.fig`

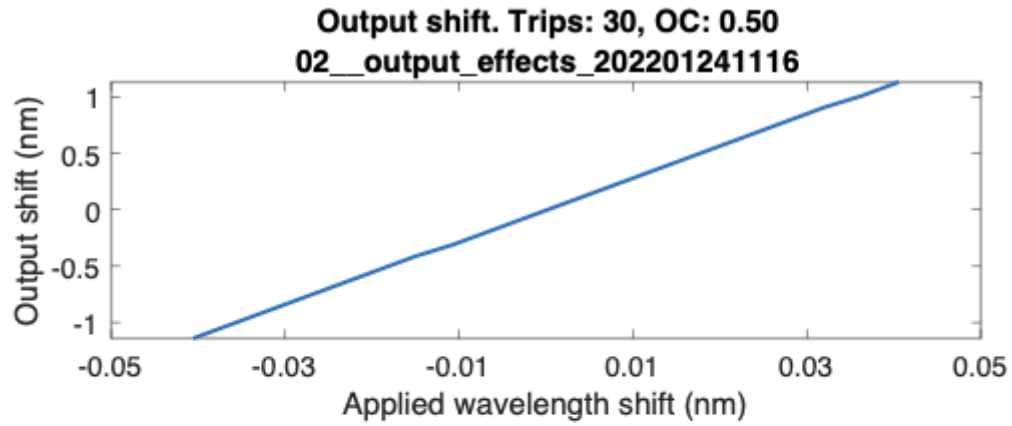


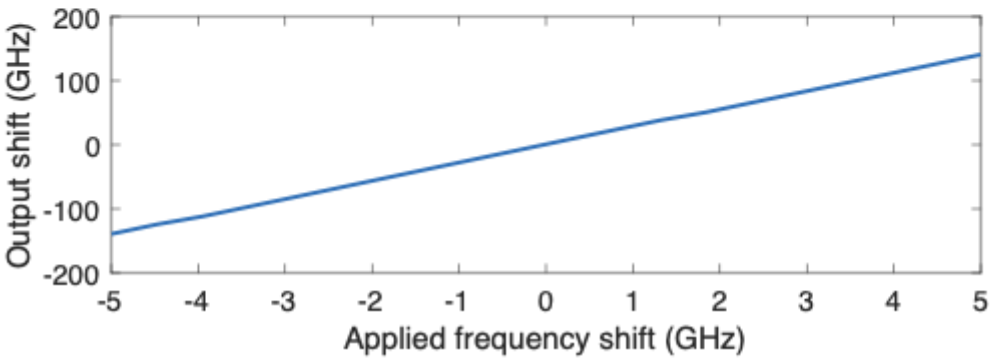
- The soliton doesn't form at all; the spectrum is asymmetric
- ~~Could be because once we shift the peak of the spectrum away from the 1550 nm, the Taylor expansion of $\beta(\omega)$ about ω_0 no longer holds and we don't get the same dispersion effects~~
    - No - it is just because we are "hitting it too hard" - the nonlinear nature can't really respond

A.2.3 Output Shift vs Applied Shift

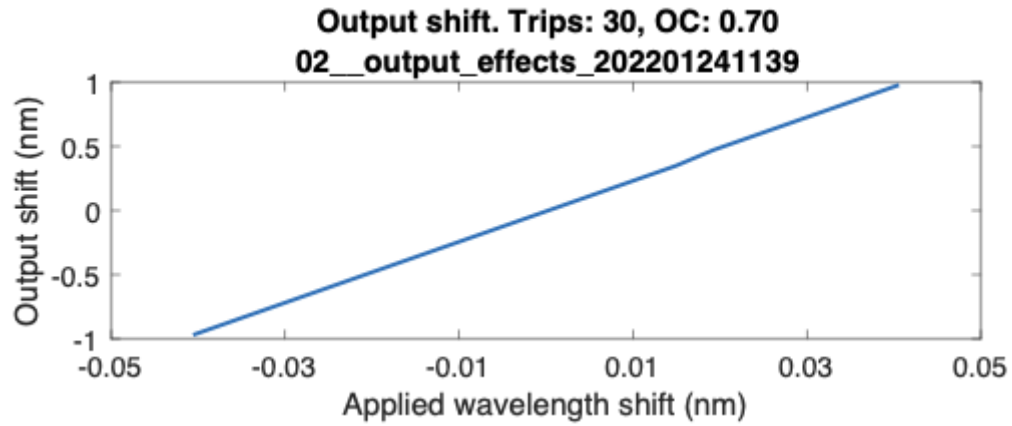
Wrote a loop in `output_effects_220121` that computes the peak wavelength of the laser output after 30 round trips, at different input frequency shifts.

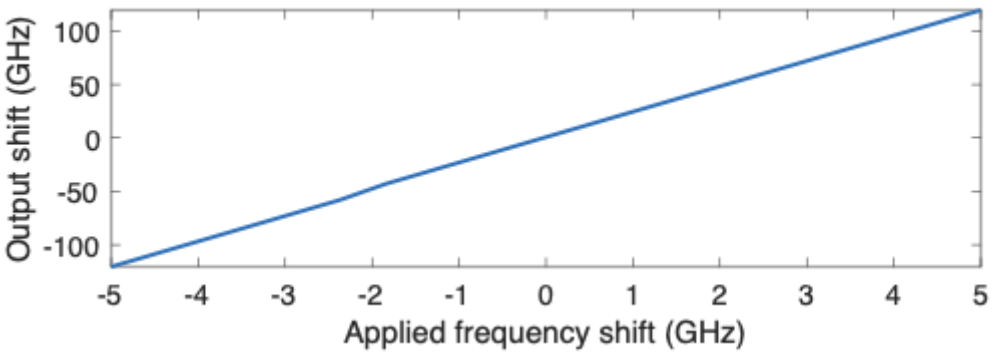
`02__output_effects__202201241116`



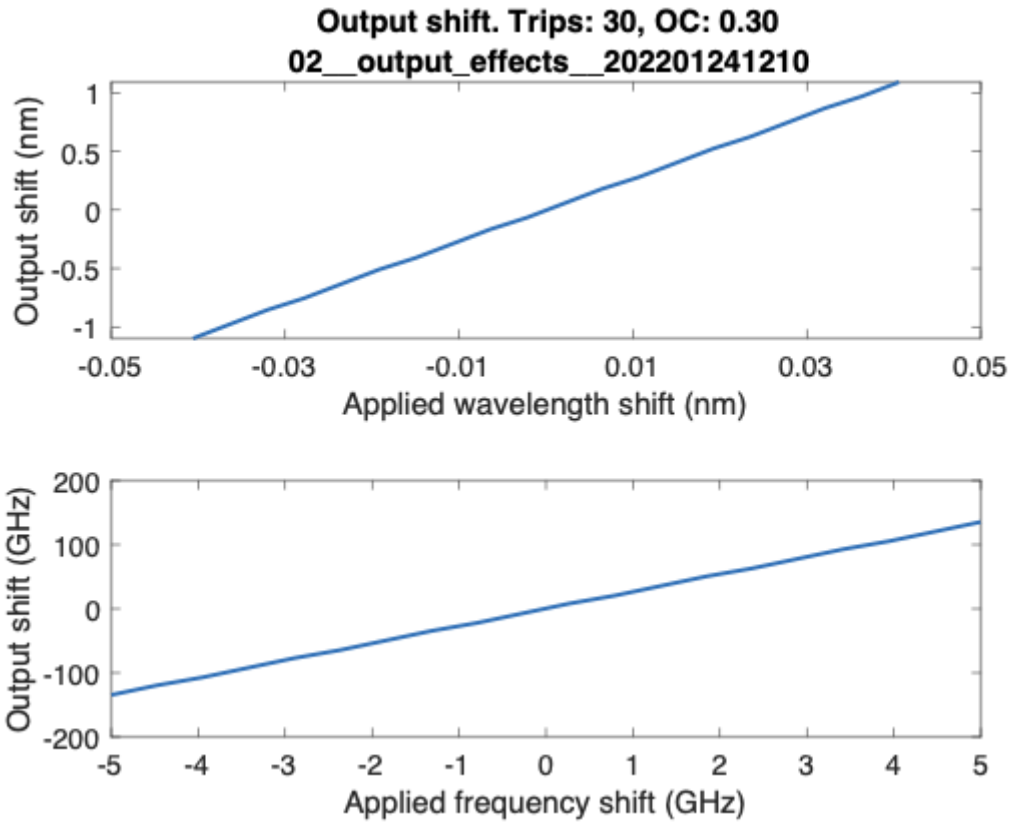



```
02__output_effects__202201241139
```





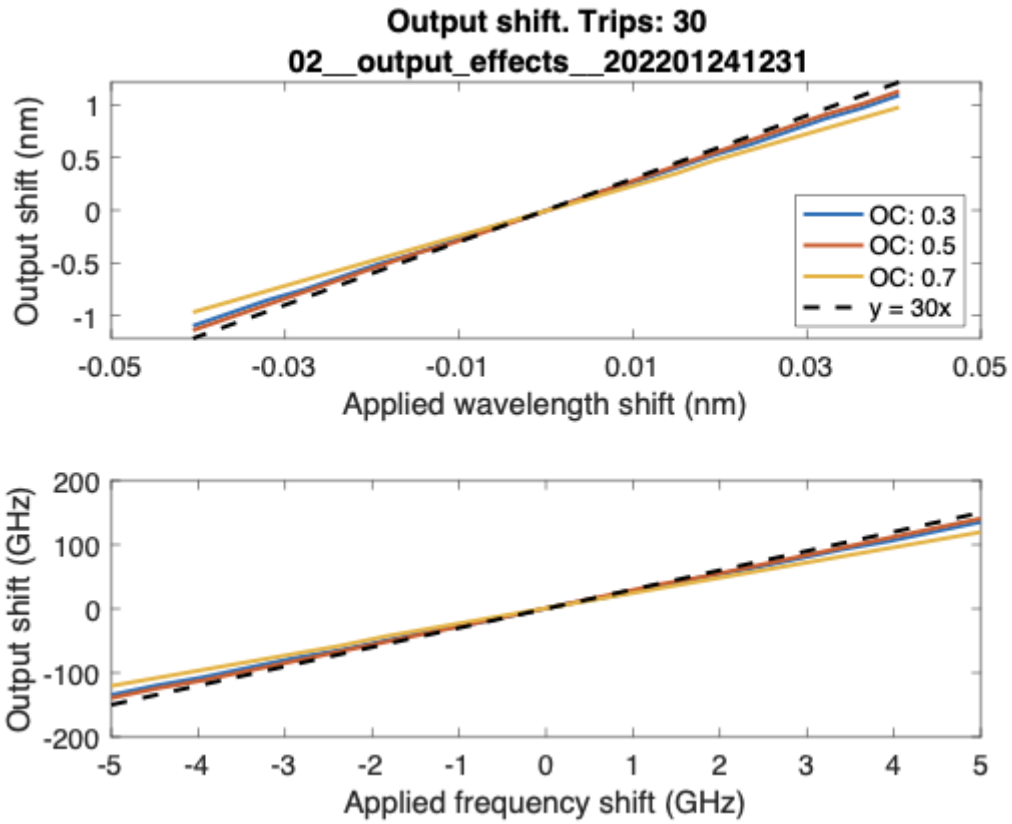

```
02__output_effects__202201241210
```



- Where `OC` refers to `feedback = 0.7` etc.
  - So 70% remains in the loop at each cycle

A.2.3.1 Comparison

```
02__output_effects__202201241231
```



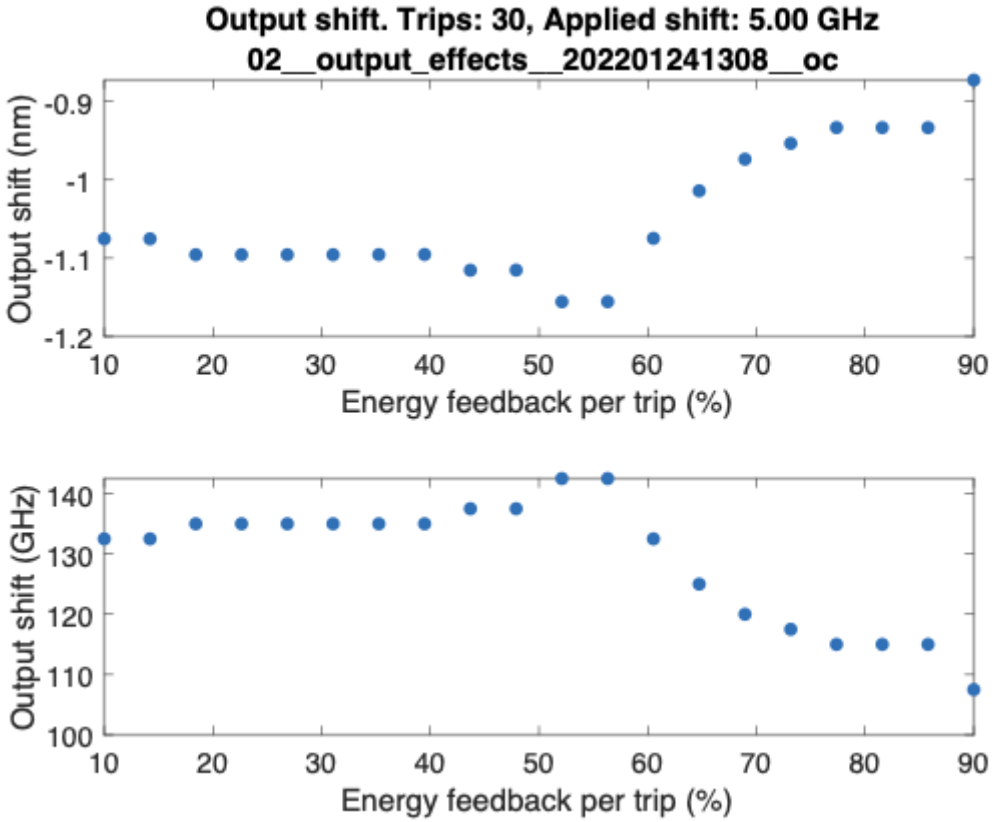
- All of the different OCs have an output that is less of a shift than if we just multiplied the applied shift by 30 (the number of round trips)
- But there is not a substantial difference between the lines, and they are all roughly linear
- So this discrepancy could be more due to the fact that the gain medium selectively applies gain centred on the original wavelength, and so the way the field is increased will be slightly different
    - Not due to the fact that photons are leaving after only having passed through the loop twice

- [ ] How does the code account for the fact that photons are leaving? Isn't the field just being decreased by `E = sqrt(feedback).*E`?

A.2.3.2 Output Coupling

Varied the `feedback` variable which controls the % of electric field amplitude square (or % of energy) that remains in the loop after each round trip.

> `02__output_effects__202201241308__oc`



- There are variations in the output shifts, with the maximum shift occurring near 0.55 (so 45% of energy leaves)
    - But there is still very minimal change at different output coupling percentages

A.3 Outcomes

- The gain profile has a bandwidth of 50 nm
    - So there is still a way to go before we see saturation (where the soliton can no longer be shifted in wavelength)
    - However, the shifts we need are only really ~ 1 nm so this should be okay
    - But we still want to see how far we can push it and what output coupling is best

A.4 Questions

- [x] ~~How do we measure the output effects?~~
    - ~~Is that different to what is plotted (i.e. the field in the fibre)?~~
        - ~~Won't it just be a portion of that field~~
    - ~~Yes, just look at the shift of the peak of the spectrum~~

- [x] ~~Can I have access to *Absence of Galilean invariance for pure-quartic solitons* or other relevant papers for this section?~~
    - ~~Yes, Martin will send~~
- [x] ~~How do I input quadratic dispersion into the program?~~
    - ~~Either get rid of the `WaveShaper` from the cycle, or input a dispersion relation that cancels out the fibre's higher order terms~~
    - ~~But those higher order terms shouldn't make too much of a difference~~
- [x] ~~How do I ensure that the pulse stays centred at t = 0?~~
    - ~~Change the `lambda0` to be the correct peak of the new soliton~~
    - ~~Run it once to get the peak and then go back and change it~~
        - ~~Didn't work~~
- [x] ~~Why isn't the output of the laser the same as the field inside the laser?~~

A.5 To Do

Using the quadratic dispersion relations:

- [x] ~~When we use the frequency shifter, what is the output frequency shift versus input shift?~~
- [x] ~~Experiment with different output coupling~~
    - [x] ~~How do we optimise the largest possible output frequency shift for a given applied shift~~
    - [x] ~~And when does it get too big and go kaput~~
- [ ] Find how many round trips it needs until it saturates (doesn't change wavelength any more)
    - Possibly reduce the `delta z step` or something
    - Or reduce the gain bandwidth a bit
- [ ] Then re-run the plots above, going until saturation so that it is no longer linear
- [ ] Check with Antoine how I can speed up the code?

Eventually we want to move on to the quartic dispersion, but we need to understand this quadratic first.

B Analytic Project

Aims

- [ ] Redo the derivations and continue working on the third order substitution to the nonlinear ODE

B.1 Notes

Revised the process for how to solve the nonlinear DE:

Process

! Step 3 is incorrect. See Logbook_03_220124 for the correct approach !

Equation (1) is of the form

$$L[u] + N[u] = 0$$

where L is a linear function, and N is nonlinear ($x \mapsto x^3$)

1. We start finding solutions by taking $L\left[u^{(0)}\right] = 0$, only including sech^1 terms
    - There could be sech^3 terms but we ignore them for now and deal with them in the third order case
2. Then let the third order function $u^{(1)}$ be defined as per Equation (3)
3. Use $L\left[u^{(1)}\right] + N\left[u^{(0)}\right] = 0$, only including sech^3 terms, to write the coefficients C_k in terms of A, B etc.
4. We can then repeat this process for higher orders.

- This is not the only possible approach, but it is a good starting point

B.2 Results

B.2.1 Mathematica

First attempt: `01_substitution_220117.nb`
Cleaned up and correct: `01_substitution_220120.nb`

B.2.1.1 First attempt

Initially tried to write a function that would automatically simplify the derivatives etc. into a "nice" form where I could easily read off the terms, determine the scaling, and then equate coefficients.

- Didn't manage to get a nice solution, in particular because it kept converting trig into double angles, using tanh and coth when I want just sech and csch
- Tried to just use `csch` instead of `Csch[x sigma]` but that still didn't perform as nicely as I wanted, since it wasn't then able to use trig identities to simplify e.g. csch^2 into sech^2
- Perhaps there is a way to get this working using better complexity functions for `FullSimplify`, but I wasn't able to do it yet

Instead, just went a more manual approach where I would read off the coefficients of powers of `sech` etc., and then combine them more manually into simplified forms

- The process ended up being:

1. Compute the expression fully
2. Group the coefficients by $\operatorname{sech}(x\sigma), \operatorname{csch}(x\sigma), \operatorname{sech}(x\sigma)$
    - Had to group twice because it didn't recognise them all the first time
3. Then read off the coefficients that produce a product of order ≤ 3
4. Combine like terms into one equation for sech^3, one for $\operatorname{sech}^2 \operatorname{csch}$ etc.
    - All of the terms ended up being of order 3, as expected

> **Expected**
>
> This was expected because when I did the *simplification* of setting $\cos = \sin = 1$ etc., I found that sech^3 was the lowest order present in the expression.

5. Convert $\tanh = \text{sech} / \text{csch}$ etc
    - This had to be done by using variables called `sech` instead of the actual functions, since it seemed to automatically convert them back into `Tanh` etc.
6. Use $\cosh^2 - \sinh^2 = 1$ to reduce the number of terms
    - Made use of `Together` to put it all on one fraction, so that we don't need to worry about denominators only the numerator
    - We can also drop the 1 term, since that will only contribute to higher order coefficients
    - Ended up with just two terms: $\cosh / \sinh^4$ and $1 / \sinh^3$
7. For the two coefficients, then use $\cos^2 = 1 - \sin^2$ etc. to simplify the trig terms, and group them together
    - This gave a set of four equations, with no x dependancy, that must all be 0
8. Solve these four equations simultaneously (using `Solve[]`)

B.2.1.2 Final attempt

I then went through and cleaned up the code, simplified the process. The final steps were

1. Convert the trig functions into variables, `Cosh[_] -> cos` etc., to stop Mathematica from simplifying
2. Place the expression onto a single fraction and extract the numerator, so that we can more easily compare coefficients
    - The denominator was $\cosh^{-7} \sinh^{-7}$
3. Use $\cosh^2 - \sinh^2 = 1$ to convert $\cosh^k$ into $\sinh$ terms (except for odd powers)
4. Extract the relevant coefficients ($\cosh \sinh^1 0$ and $\sinh^{11}$)
5. Simplify the trig terms using $\cos^2 = 1 - \sin^2$ etc.
6. Equate the coefficients together to $= 0$ to create 4 simultaneous equations
7. Solve these equations using `Solve`

Converted this into a script function that can be used: `PQSHelper.wl`

B.2.1.3 Results

Incorrect

These results here are incorrect, they have been corrected in Logbook_03_220124

~~What I found was that all of the coefficients involved functions of x, e.g. $c_3 \cos(\sigma x)^3$ ~~

- ~~These cannot be identically zero for all x ~~
  - ~~Only one of these rows actually involves A, B - so changing the values will not affect this either. ~~
  - ~~Even if all $c_i = 0$, the terms involving A, B don't involve c_i, so those will be nonzero~~
  - ~~It will only be identically zero if A, B, c_i all are 0 (as expected)~~
- ~~Therefore it appears that it is not possible for this function to be 0 up to order 3 ~~
  - This was solved by using trig identities to simplify the $\sinh$ terms further

$$\begin{pmatrix} 3A^2B\Gamma + 72c(0)\sigma^4 - 232c(1)\sigma^4 + 24c(2)\sigma^4 + 264c(3)\sigma^4 \\ -3A^2B\Gamma + B^3\Gamma + 320c(1)\sigma^4 - 320c(3)\sigma^4 \\ 3AB^2\Gamma + 264c(0)\sigma^4 - 24c(1)\sigma^4 - 232c(2)\sigma^4 - 72c(3)\sigma^4 \\ A^3\Gamma - 3AB^2\Gamma - 320c(0)\sigma^4 + 320c(2)\sigma^4 \end{pmatrix}$$

Solving these equations, we get

$$\begin{aligned} c_0 &= -\left(2A^3 + 9A^2B + 6AB^2 + 9B^3\right)\varphi \\ c_1 &= 3\left(3A^3 + 2A^2B + 3AB^2 - 2B^3\right)\varphi \\ c_2 &= -3\left(2A^3 + 3A^2B - 2AB^2 + 3B^3\right)\varphi \\ c_3 &= \left(9A^3 - 6A^2B + 9AB^2 - 2B^3\right)\varphi \end{aligned}, \qquad \varphi := \frac{\Gamma}{1280\sigma^4}$$

Substituting in the guesses

> His current best estimates for A and B are $A = 0.4384$ and $B = 1.0170$.

we get

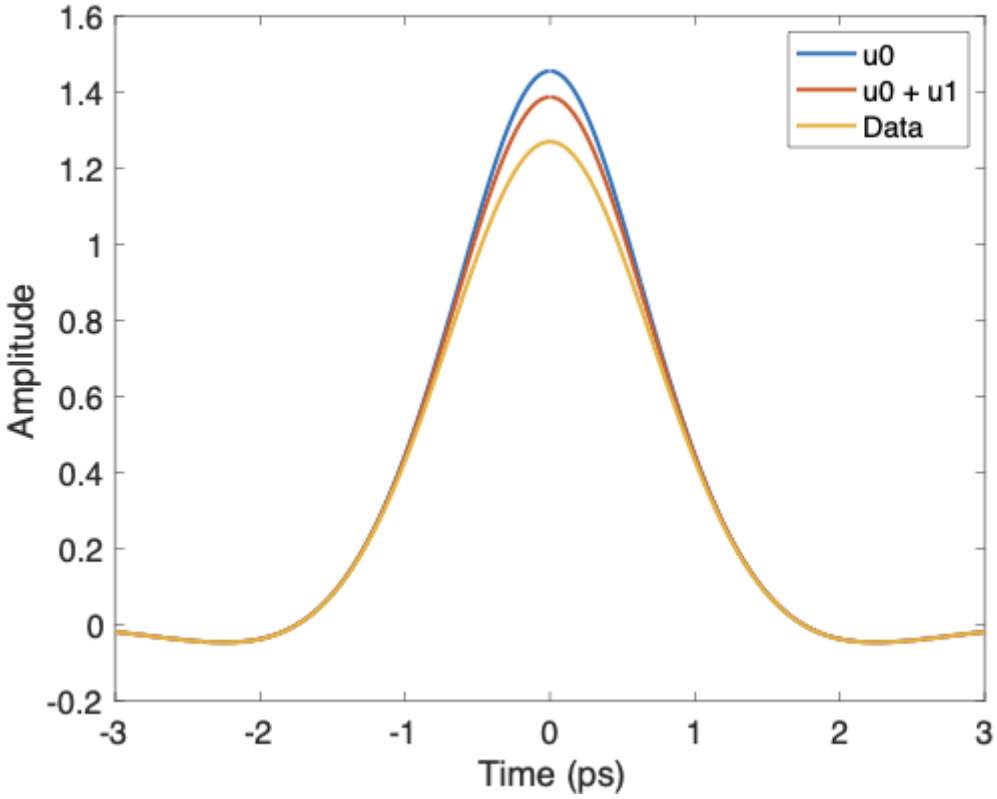
$$c_0 = -0.0441097,\ c_1 = 0.000935146$$
$$c_2 = -0.0281592,\ c_3 = 0.00488343$$

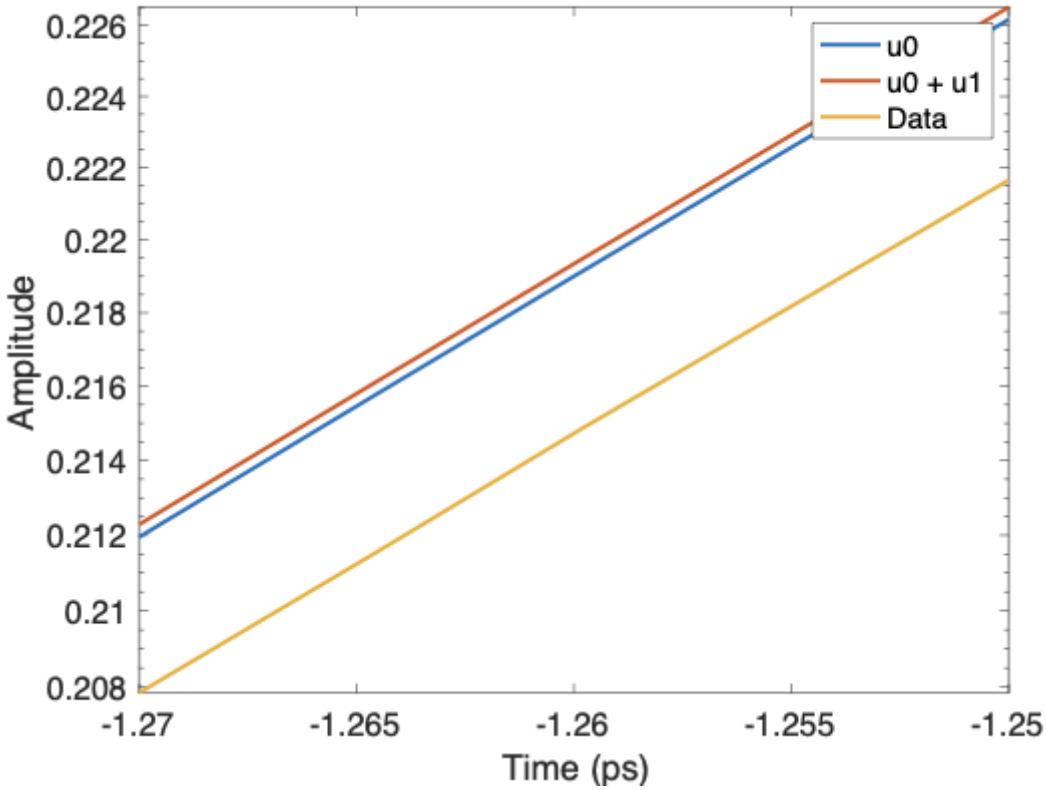
B.2.1.4 Note

- Also repeated this using $u^{(0)}$, and we get that the relevant terms are identically 0, as expected
  - So this method / approach does seem to work
- The final values we get do give 0 for the third order terms when substituted into the (simplified) expression (using $\cosh = \sinh$ etc.)
- When I first did the code, I also realised I made a mistake and so I had $u^{(0)} = A\frac{\cos}{\cosh} + B\frac{\cos}{\cosh}$ instead of $B\frac{\sin}{\sinh}$
  - This gave much simpler results for the final coefficients (because $A \mapsto A + B, B \mapsto 0$ effectively)

B.2.2 Simulations

In MATLAB, using the data given by Long, we can compare the estimates





B.3 Outcomes

- It is possible to write down $u^{(1)}$ such that it satisfies our equations

B.4 To Do

- [x] ~~Check if these results hold for arbitrary A, B~~
- [x] ~~Make sure that this approach works for the $u^{(0)}$ case too~~
- [x] ~~Check with Martijn what to do next~~
  - [x] ~~Simplify the terms more using $\sinh^2 = -1 - \cosh^2$ etc~~
    - ~~There should only be 4 or 5 coefficient / expressions in the final simplification~~
- [x] ~~Compare our solution to the data given from numerically solving the system~~
  - [x] ~~Compare the fit of the new solution in the tails using a log scale~~
    - ~~Not too sure how to do this~~
- [x] ~~Tidy up the Mathematica working out to show to Martijn~~

B.5 Questions

- [x] ~~How do we plot log scale when there are negative numbers?~~
  - ~~Just have the lines go off to infinity, we are only looking at the position of the bumps~~
- [x] ~~Is $\Gamma < 0$ or > 0? There is a discrepancy between the Tutorial ODE and the ODE in your working out~~
  - ~~Martijn says it should be $-\Gamma u^3$ as per the tutorial paper~~
  - [x] ~~Check with Long to see which one he used~~
    - ~~He used $-\Gamma u^3$~~
- [x] ~~For Monday: Discuss finding A, B using Mathematica rather than just visually matching~~
  - ~~We know the decay rate and oscillations in the tails (encoded in the σ parameter)~~
  - ~~Write $\mathcal{A}\left(\cos\varphi\frac{\cos}{\cosh} + \sin\varphi\frac{\sin}{\sinh}\right)$ instead of A, B~~
    - ~~i.e. mod arg form~~
    - ~~Or maybe $\cos^2\varphi$ etc.~~
  - ~~Vary φ and $\mathcal{A}$ to visually match with the numerical result~~
    - ~~Subtract my result from the numerical result (on a log scale)~~
    - ~~View on a log scale so that the tails are linearly decreasing (or straight lines once I subtract)~~
      - ~~Look very far into the tails (like 1e-9) so that the nonlinear term is very small~~
  - ~~Then add the cubic terms to the solution ($u^{(1)}$) and see how it works~~

C Other

C.1 Questions

- [x] ~~In Figure 14 of the tutorial, so when the pulse is not in the WS, does it behave like a conventional soliton with just quadratic and some cubic dispersion?~~
    - ~~Instantaneously, it is not a solution. It is breathing (getting broader and narrower). On average though, it has dispersion that is quartic and negative. It is an **effective soliton**.~~
- [x] ~~What am I presenting to the group on Friday?~~
    - ~~Talk for five minutes about what I am doing and introducing myself. Have some slides prepared~~