Tags: #logbook - Denison
Links:
Files: `01_substituion_220115.nb`, `PureQuartic.csv`, `FreqShift.m`, `cavity_PQS_August2019.m`

Logbook_220115

From Saturday 15th to Monday 17th January

A Numeric Project

Aims

- [ ] Play with the code, keeping the pure quadratic dispersion
- [ ] Add in the frequency shifter and see what happens

A.1 Notes

- The dispersion relation becomes quadratic when we shift into the moving frame, because the group velocity is zero (the slope)
  - It will be an upside down parabola, as we take $\beta_2 < 0$ to create the solitons
- The Fourier transform of a soliton doesn't lie on the dispersion relation. It is a straight line above it
  - Therefore it is nonlinear
- When a soliton emerges it is created in the frame so that the height of the 3 peaks are the same for the dispersion relation
  - This is quite nontrivial why this is so
  - Might be due to complicated phase structures that arise in these other solitons
- "A quartic curve is not Galilean invariant" - there is a unique frequency where β_2 and β_3 are zero. If we shift into another frame, they will be nonzero

    - It will no longer be quartic and symmetric
    - In contrast, a quadratic curve is invariant - it will still be a parabola when we shift frames
- What if we used a frequency shifter in the laser cavity to "nudge" the soliton towards a different profile?

In systems such as a PQS laser, we've observed that there is a preference for PQS to form at a particular frequency and particular group velocity. But these are part of a larger family of solitons, travelling at different speeds and with more complex phase structures and asymmetric spectra.

So the question is can shift the frequency of the soliton as it is created (such as from an acousto-optic modulator) to nudge the soliton into these other forms? In particular we want to see how the magnitude of the applied shift affects the output of the laser, and how it interplays with other parameters such as coupling.

A.1.1 MATLAB Code

The entry point for the code is `cavity_PQS_August2019.m`

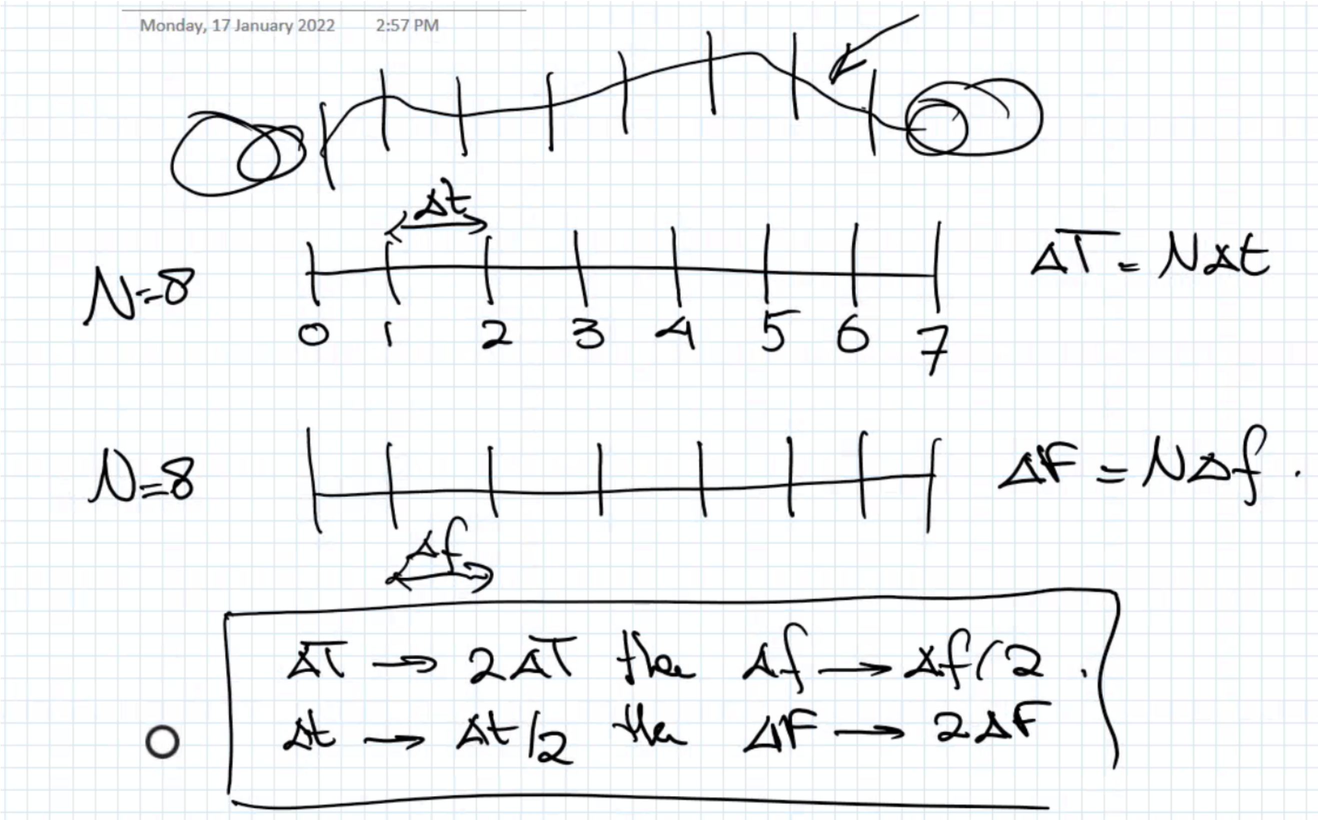
- Starts with a random pulse and certain parameters, and then evolves the pulse over 100 round trips
- The components include:
    - `SMF`: the propagation in the typical single mode fibre
    - `SA`: saturate absorber which gives us the short pulse widths
    - `gain_PQS`: propagation in the gain medium
    - `WaveShaper`: the screen which adjusts the phase of the soliton
    - `OC`: energy lost due to the output coupler
- While doing this, energy is being lost due to power loss in connectors between sections
- This process is called breathing

To input quadratic dispersion, I give the following to the WaveShaper:

```
% beta coeff for WS, for quadratic dispersion
beta2_WS = 0;               % beta2  (s^2/m)
beta3_WS = -0.12e-39;       % beta3 with opposite sign for Wavesha
beta4_WS = 2.2e-54;         % beta4 with opposite sign for Wavesha
```

such that the dispersion from the SMF is cancelled for β_3, β_4, only leaving β_2 nonzero.

A.1.2 Fourier Transforms

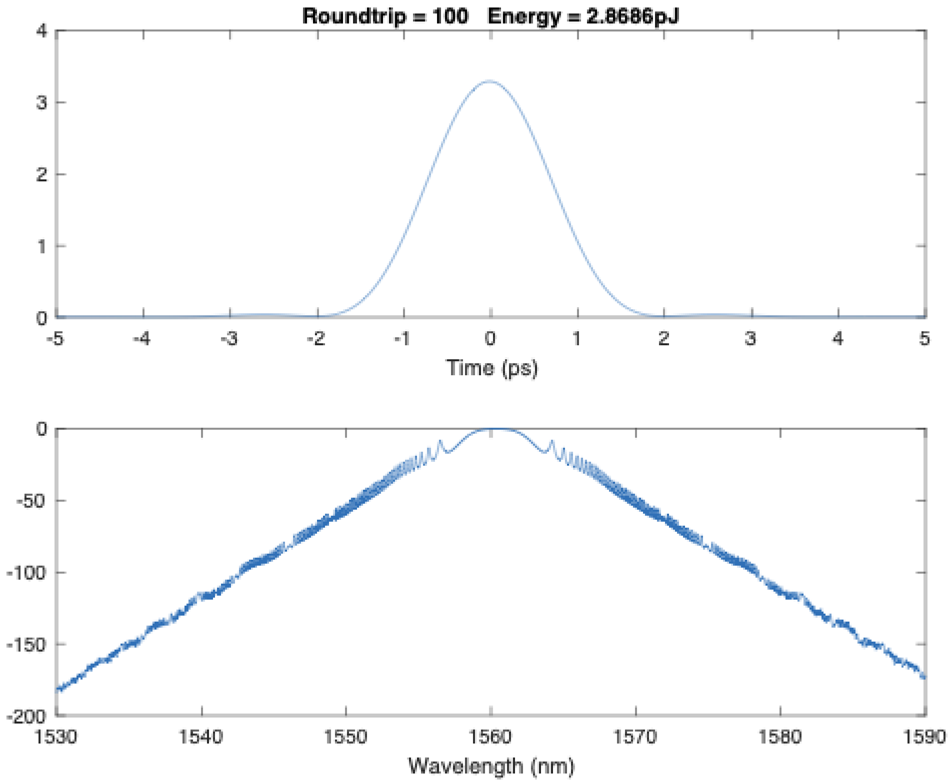


A.2 Results

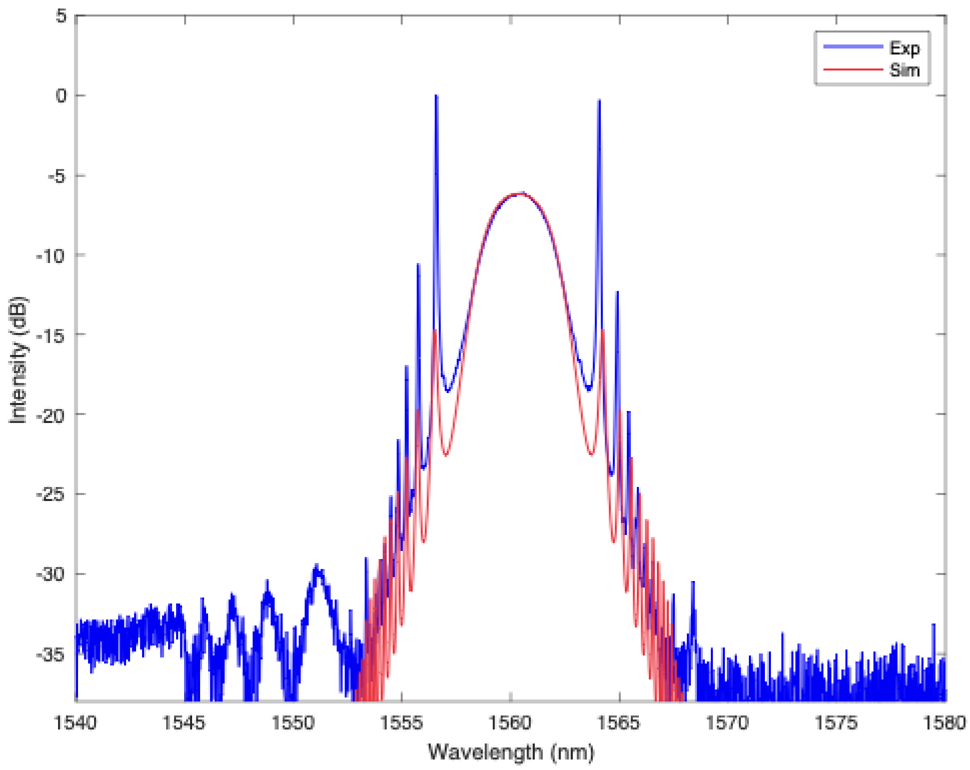
A.2.1 Unmodified

Running the code (unmodified) produces three figures:

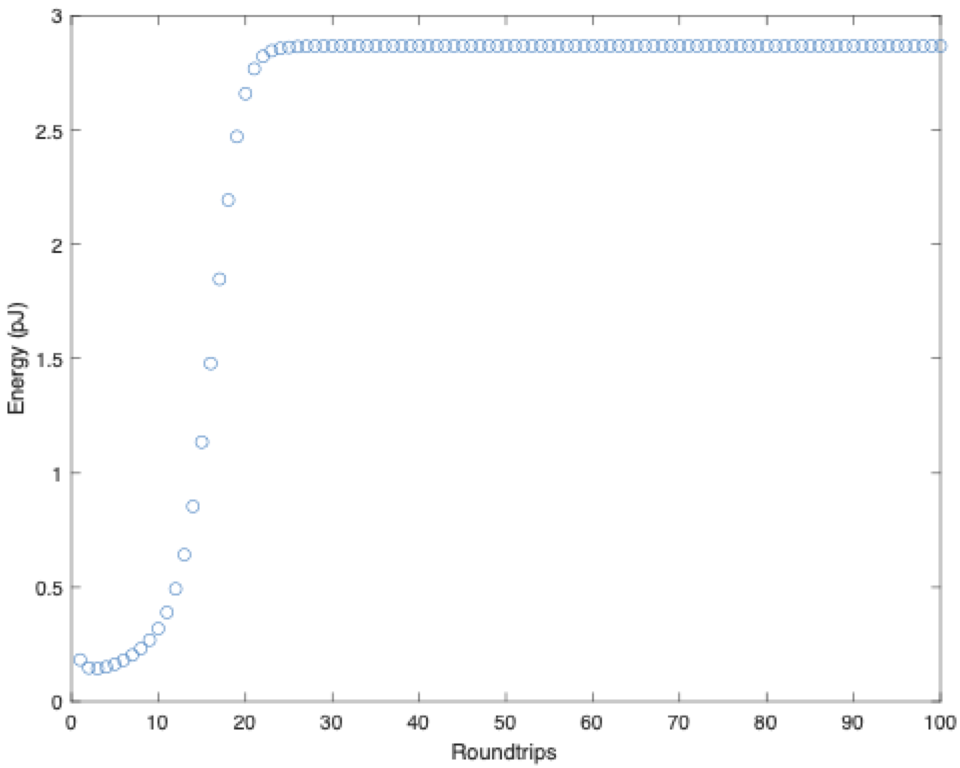
1. Soliton temporal profile and spectrum, evolving with each round trip
   - For the original set up, we would expect symmetry in the soliton and the spectrum is centred about the input pulse wavelength: 1560.35×10^{-9}



2. Comparison of the profile from the experimental laser output (`Exp`) and the simulated value (`Sim`)
- Relatively close profile match, with smaller amplitude



3. The energy of the pulse at each round trip
- It plateaus because of the way laser gain works, can become saturated etc.



A.2.2 Frequency Shifter

We would expect to see that the pulse shape is not strongly affected, but the wavelengths are shifted (to the left, for a positive frequency shift).

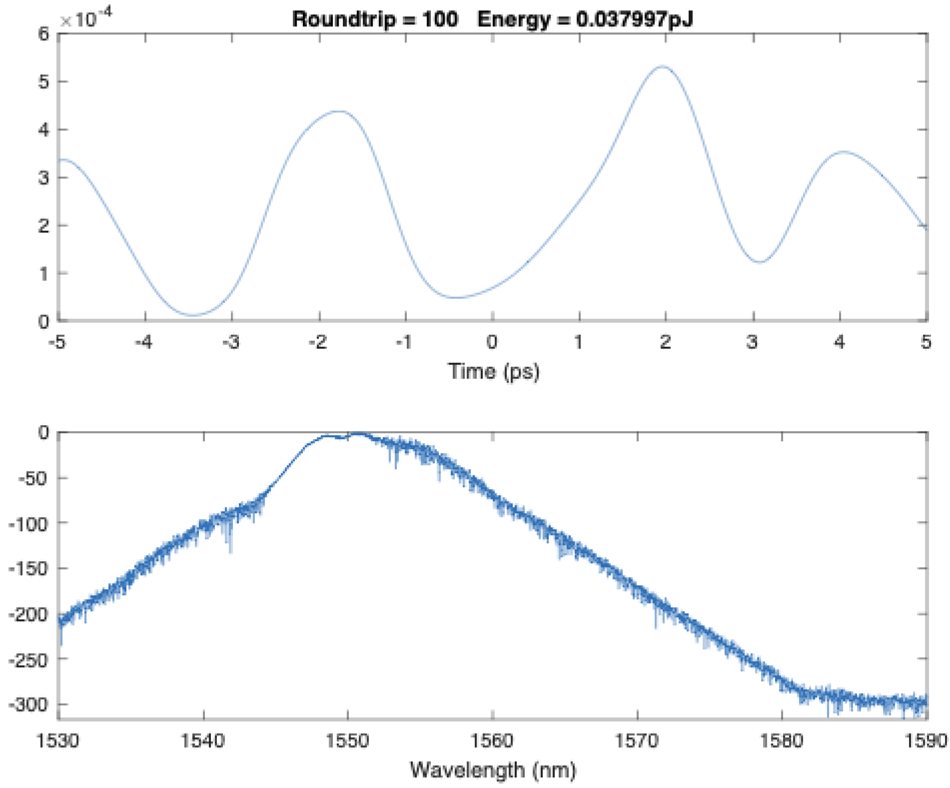
Created file `FreqShift.m`

```
% Get the spectrum
TE = fftshift(ifft(fftshift(E)));
% Shift the spectrum by a given frequency shift
shift = round(freqdelta / df);
TE = [zeros(1, shift) TE(1:end-shift)];
% Recompute the field
E = fftshift(fft(fftshift(TE)));
```

- Currently have just placed it after the `WaveShaper` function call in the round trip loop
- `df = 2.5e9` is the frequency resolution in the simulation

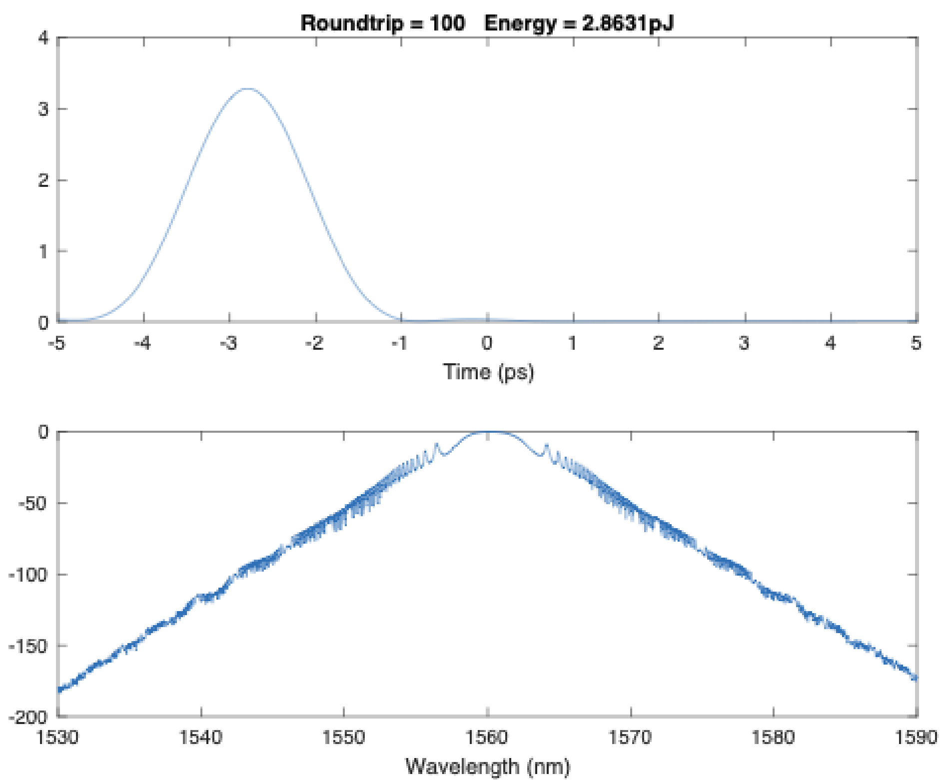
Using `freqdelta = 10*df`, we get:

- The wavelengths have shifted as expected, but no soliton emerges



- The amplitude is ~ 0

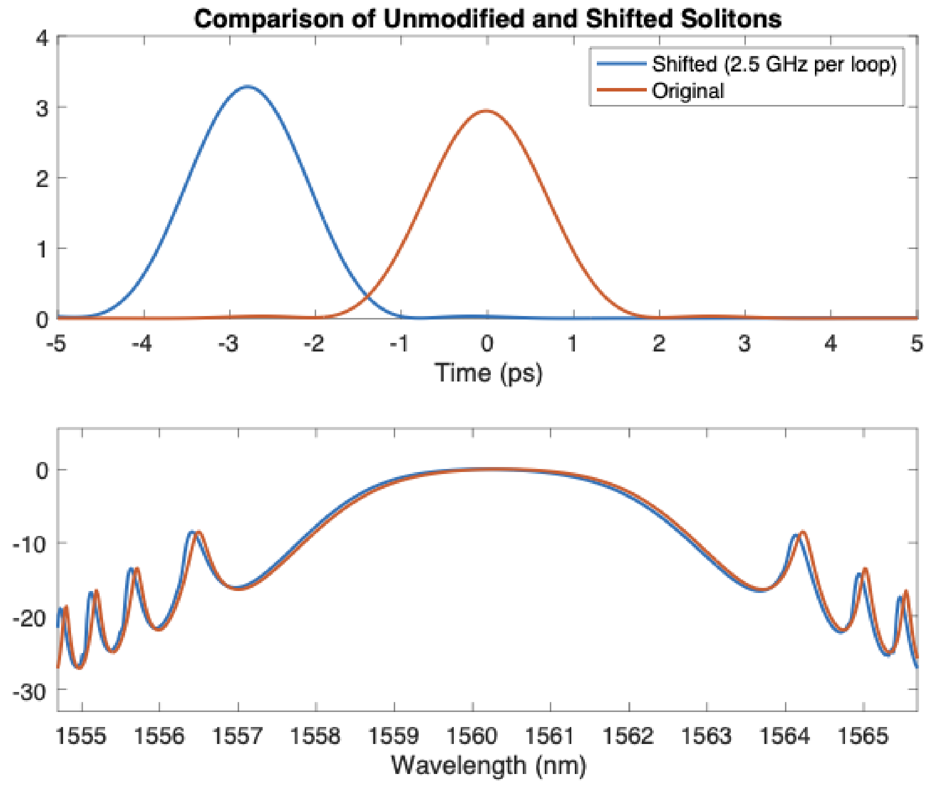
Using `freqdelta = df`: (i.e. the minimum frequency interval)

> `01__202201171130.fig`
>
> 
>

- The soliton has shifted backwards temporally, but kept its shape and energy
- Wavelengths look similar to the unmodified soliton

Comparing the original and the shifted (`1df`) solitons:

`01__202201171146.fig`



- The wavelengths are slightly shifted to the left (as expected)

> **Explanation**
>
> It makes sense that the pulse has shifted earlier in time because negative β_2 means that $\frac{\mathrm{d}\beta}{\mathrm{d}\omega} < 0$ in the moving frame, or just a smaller $\frac{\mathrm{d}\beta}{\mathrm{d}\omega}$ in the stationary frame. This means that we have a negative **inverse** group velocity (moving) or smaller (stationary), and so the pulse travels backwards (moving) or faster (stationary).
>
> Recall that $\frac{1}{v_g} = \frac{\mathrm{d}\beta}{\mathrm{d}\omega}$. A faster pulse means anomalous dispersion.

A.3 Outcomes

A.4 Questions

- [x] ~~What is `WaveShaper` doing?~~
- [x] ~~What is `OC` doing?~~
- [x] ~~What are the papers relevant to this topic?~~
- [x] ~~Have I implemented the frequency shifter correctly? Where in the path should I place it?~~
    - ~~Place it anywhere~~
    - ~~Multiply by $e^{-i\omega t}$ instead of doing the Fourier transform~~
- [x] ~~What is a typical frequency shift per round trip?~~
    - ~~Small, around 10 MHz using the acoustic frequency shifter (acousto-optic effect)~~

A.5 To Do

- [ ] Read the soliton paper

- [ ] Play with different frequency shifts and different positions of the component in the cycle

- [ ] Check by how much the output has changed
    - [ ] Figure out what this means??
    - [ ] See to what degree the process remains the same when playing with the output coupling and the frequency shift

---

B Analytic Project

Aims

- [ ] Create the Mathematica script to evaluate the derivatives
- [ ] Check Martijn's results for the lowest order case
- [ ] Produce results for the next case

B.1 Notes

The main ODE we are considering is:

$$u'''' + 4\sigma^4 u + \Gamma u^3 = 0 \tag{Eq. 1}$$

where

- $\sigma^4 := \dfrac{6\mu}{|\beta_4|}$
- $\Gamma := \dfrac{24\gamma}{|\beta_4|}$
  This is chosen so that we have solutions of the form
  $$u^{(0)} = Ae^{\sigma t}\left(e^{i\sigma t} + e^{-i\sigma t}\right) = Ae^{(1+i)\sigma t} + Ae^{(1-i)\sigma t}$$
- But this approach doesn't work, since when taking higher orders the series does not converge
  - Since we get terms like $e^{3\sigma t}$, $e^{5\sigma t}$ which will not converge at large t

We instead want to work with the functions (which hopefully may be able to converge)

$$u^{(0)} = A\frac{\cos\sigma x}{\cosh\sigma x} + B\frac{\sin\sigma x}{\sinh\sigma x} \tag{Eq. 2}$$

$$u^{(1)} = \sum_{k=0}^{3} C_k \left(\frac{\cos\sigma t}{\cosh\sigma t}\right)^{(3-k)} \left(\frac{\sin\sigma t}{\sinh\sigma t}\right)^{k} \tag{Eq. 3}$$

- Martijn's initial calculations showed that the $u^{(0)}$ substitution satisfies Eq 1. to the lowest order
  - i.e. to the order that decreases as $\sinh^{-1}\sigma t$ and $\cosh^{-1}\sigma t$

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.

My aim is to check if Step 3 is possible using our functional form of $\cos / \cosh$ etc.

B.1.1 Pure Quartic Soliton Solution

Long has provided me with a pure quartic soliton solution, `PureQuartic.csv`

- With $\beta_4 = -1$ (dispersion term); $\mu = 1$ (nonlinear phase shift); $\gamma = 1$
  - This means that $\sigma^4 = 6$ and $\Gamma = 24$
- Time in the first column, amplitude in the second column

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

B.2 Results

THIS IS INCORRECT:
The process of generating solutions to Equation (1) is:

1. Choose the first ansatz $u^{(0)}$

2. Then set $\Delta_0 := u^{(1)} - u^{(0)}$ to be of the form $u^{(0)''''} + 4\sigma^4 u^{(0)} + {u^{(0)}}^3$ (i.e. the left over parts when substituted into the differential equation)
3. Solve for the constants in $u^{(1)}$ by letting $(\Delta_0)'''' + 4\sigma^4 (\Delta_0) + \Gamma {u^{(0)}}^3 = 0$
4. Then set $\Delta_1 := u^{(2)} - u^{(1)}$ to be of the form ${u^{(1)}}^3$ and repeat

When we take this approach with the ansatz in Equation (2), then we find that the DE is satisfied to the lowest order i.e. that which decreases as $\sinh^{-1}$ and $\cosh^{-1}$.

B.2.1 Mathematica

- Created `01_substituion_220115.nb`
  - Implemented the ODE and ansatz

```
CC[x_] := Cos[\[Sigma] x] / Cosh[\[Sigma] x];
SS[x_] := Sin[\[Sigma] x] / Sinh[\[Sigma] x];

ODE[f_] :=
  D[f[x], {x, 4}] + 4 \[Sigma]^4 f[x] + \[CapitalGamma] f[x]^3;

u0[x_] := a (CC[x]) + b (SS[x])
```

- Checked the calculations from Martijn, they are correct for the lowest order ansatz
- To consider the scaling when substituted into the ODE, initially used the `Limit` command to take `x -> Infinity`, however that was slow and not always conclusive when dealing with the higher powers and higher order ansatz
- Instead, I simplified the expression, so that we can see the scalings

```
Simplification = {Tanh[\[Sigma] x] -> 1, Coth[\[Sigma] x] -> 1,
 Csch[\[Sigma] x] -> Sech[\[Sigma] x], Cos[\[Sigma] x] -> 1,
 Sin[\[Sigma] x] -> 1}
```

- This give

```
In[102]:= Collect[ODE[u0] /. Simplification, Sech[σ x]]
```

Out[102]= $\left(A^3 \Gamma + 3 A^2 B \Gamma + 3 A B^2 \Gamma + B^3 \Gamma - 32 A \sigma^4 - 8 B \sigma^4\right) \mathrm{Sech}[x\,\sigma]^3 + \left(5 A \sigma^4 + 5 B \sigma^4\right) \mathrm{Sech}[x\,\sigma]^5$

We see that the u0 solution scales with 1/cosh(x)^3

- With the next order ODE ($u^{(1)}$), we got a similar scaling:

```
In[104]:= Collect[ODE[u1] /. Simplification, Sech[σ x]]
```

Out[104]= $\left(352 c \sigma^4 - 192 d \sigma^4 - 96 e \sigma^4 + 64 f \sigma^4\right) \mathrm{Sech}[x\,\sigma]^3 + \left(-696 c \sigma^4 - 96 d \sigma^4 + 8 e \sigma^4 - 96 f \sigma^4\right) \mathrm{Sech}[x\,\sigma]^5 + \left(33 c \sigma^4 + 9 d \sigma^4 + 9 e \sigma^4 + 33 f \sigma^4\right) \mathrm{Sech}[x\,\sigma]^7 + \left(c^3 \Gamma + 3 c^2 d \Gamma + 3 c d^2 \Gamma + d^3 \Gamma + 3 c^2 e \Gamma + 6 c d e \Gamma + 3 d^2 e \Gamma + 3 c e^2 \Gamma + 3 d e^2 \Gamma + e^3 \Gamma + 3 c^2 f \Gamma + 6 c d f \Gamma + 3 d^2 f \Gamma + 6 c e f \Gamma + 6 d e f \Gamma + 3 e^2 f \Gamma + 3 c f^2 \Gamma + 3 d f^2 \Gamma + 3 e f^2 \Gamma + f^3 \Gamma\right) \mathrm{Sech}[x\,\sigma]^9$

```
In[132]:= %104 /. ((# → 1) & /@ {c, d, e, f})
```

Out[132]= $128 \sigma^4 \mathrm{Sech}[x\,\sigma]^3 - 880 \sigma^4 \mathrm{Sech}[x\,\sigma]^5 + 84 \sigma^4 \mathrm{Sech}[x\,\sigma]^7 + 64 \Gamma \mathrm{Sech}[x\,\sigma]^9$

So this also scales with 1/cosh^3

- Finally, with the second order $u^{(2)}$, we have a $1/\cosh(x)^5$ scaling
- And $u^{(3)}$ scales with $1/\cosh(x)^7$ etc.

Note that we can generate the $u^{(n)}$ using

$$u^{(n)} = \sum_{i=0}^{2n+1} a_i \left[\frac{\cos(\sigma x)}{\cosh(\sigma x)}\right]^i \left[\frac{\sin(\sigma x)}{\sinh(\sigma x)}\right]^{2n+1-i}$$

```
getScaling[n_] :=
 Collect[
  ODE[x |->
     Sum[coef[i] CC[x]^i SS[x]^((2 n + 1) - i), {i, 0, 2 n + 1}] ]
   Simplification, Sech[\[Sigma] x]]
```

B.2.1.1 Outcomes

1. $u^{(0)}$ and $u^{(1)}$ have a remainder that scales with $1/\cosh(x)^3$ when substituted into the linear component of the ODE. $u^{(2)}$ has a remainder that scales with $1/\cosh(x)^5$, etc. This continues with higher $u^{(n)}$.
    - So it appears that it does converge as we want

Note: This is not really correct, see the next Logbook_02_220117 for the better interpretations.

B.3 Questions

- [x] ~~Is my method correct for generating solutions to Equation 1? What is the line with purple arrow doing in the working out?~~
    - ~~No, I should be doing something slightly different~~
- [x] ~~Is my method correct for determining the scalings with the full function?~~
    - ~~Not required anymore, we are looking at something slightly different~~
- [x] ~~What does *pure quartic soliton solution* mean?~~
    - ~~The nonlinear Schrödinger equation typically has 2nd order dispersion by assumption, but there are cubic and quartic dispersions~~
    - ~~"Pure" means that the quartic dispersion is the dominant (and only) dispersion~~
- [x] ~~How do I use Long's data? Should I be substituting in values to try minimise the remainder in the ODE?~~
    - ~~We need the values of A and B to find the successive terms~~
    - ~~When I figure out how to write my $CDEF$ coefficients, then the function that I get should hopefully be a close fit to the experimental data~~
        - ~~Better than just the $u^{(0)}$ fit~~

B.4 To Do

- [x] ~~Read the Tutorial paper~~
- [x] ~~Repeat the calculations, except following the actual process this time~~

C Other

- Worked on reading the tutorial paper

    - [ ] Can I have access to *Absence of Galilean invariance for pure-quartic solitons* please