# Numerical Methods for the DNLS Equation

## Background and Galilean Transformation

1)  … the DNLS equation with dissipation

1’)  … the DNLS with dissipation.

A Galilean transformation to a frame of reference travelling with velocity, v, can be implemented through:

2)  and 

The effect of this transformation on the equation is found through constructing the exact derivative of a function such as . Given this and (2), then

3)  as well as 

Where (2) has been used to eliminate dx’. Equating multiples of dx and dt gives the transformed partial derivatives:

4)  and 

Substituting these transformed derivatives into (1’) and then dropping the primes yields:

1’’)  … the DNLS under a Galilean transformation to a reference frame moving at speed, v.

Next, Fourier transform the equation using

5)  … where B(k,t) is the Fourier Transform of b(x,t)

(*the ‘+’ sign of ‘+ikx’ in (5) is required to be consistent with the def’n of fft used by Matlab to be consistent with F(df/dx) = +ikF(f) … s.t. ifft(fft(+ik\*f)) = df/dx* )

Resulting in,

6)  … with F[g(x)] being the transform of g(x)

## The Mixed Numerical Method

We’ll (you will, I hope!) test a mixed numerical method starting with eq. (6) above replacing the Bt and the two B terms with the following

 where , so the superscript, n, is just the index referring to a time step.

Replace the B term in ‘ikvB’ with



And replace the B in the  with



These replacements are a weighted average of B across three time steps. It’s ‘weighted’ if the coefficients in front of each term add to 1.

For the first one the coefficients are ao, (1- ao)/2 … and … (1- ao)/2, which in fact do add to 1.

Substituting these into eq. (6) and doing the algebra to solve for Bn+1 gives:

7) 

With 

There are two free parameters, ao and co. You ask, ‘What values should I use for ao and co?” That is a good question. I don’t know but there is a way to find out and here it is …

I’ve uploaded my Trial 4 to the google drive. Look at the excel sheet for trials 17, 18, 19 and 20. Reproduce these trials with the same values for L, N, bright soliton parameters, Courant … and everything else. The only parameters to vary are Lva0 (which is given a value on line 110 of Call\_DNLS\_v7) and L2nd on the next line. The result to note is the number of timesteps completed in the trial which is recorded in the excel file for your trial number (mine is Trial 4) on the sheet corresponding to your run number (mine are 17, 20). The timesteps completed is recorded in the ‘Post Run Diagnostics’ in column E, row 16. It turns out time for the onset (rather the final stage) of numerical instability depends on the values chosen for ao and co.

I propose making a table and recording your results in a form something like:

**From Trial #4**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Values for ao 🡪 | | | | | |
| Values for co 🡪 |  | 0.25 | 0.50 | 0.75 | 1 | 1.25 |
| 0.25 |  |  |  |  |  |
| 0.5 | Trial # 18  #dt’s = 523 | Trial # 17  #dt’s = 541 | Trial # 19  #dt’s = 559 | Trial # 20  #dt’s = 579 | Trial # 20  #dt’s = 601 |
| 0.75 |  |  |  |  |  |
| 1 |  |  |  |  |  |

Next, I’d probably keep increasing the value of ao until the number of time steps before failure starts to decrease. Whatever that value is, I’d hold that constant, then vary the co values until a maximum is found. Yes, then from that co value, adjust the ao values. So I’m proposing a strategy involving changing one parameter at a time until any change leads to a decrease.

A fair analogy is if your somewhere on the side of Mt. St. Helens but it’s really foggy. Yes, nothing good comes from that, but work with me. Whatever direction you’re facing, start walking. If you immediately go downhill, make a 180 and then keep walking uphill until you eventually start going downhill. Then make a 90 degree turn and start walking. If you immediately go downhill, turn around 180 degrees and keep walking until you just start to go downhill. And repeat. Stopping before you reach the crater rim … or some other cliff. With a normal-ish mountain you’re pretty much guaranteed to at least reach a local maximum, but note that it might not be the tallest part of the mountain. So a maximum found by this iterative method might just be a local maximum.

Oh, it may well be that no choice of ao or co leads to any profound improvement. Also, an optimum choice for one profile (and set of parameters like, N, L, Courant) might not be optimal for another profile or parameter choices for L, N, Courant etc. But, we will know more than we do right now.

Presumably there might be a mathematical way to find the optimum values of ao and co. My experience is that often a good enough choice can be found in less time through trial and error than reading some dense papers and grinding through the math. Which is fun if one has a lot of time.