Figuring out Matlab's FFT/IFFT and their scaling

The FFT/IFFT in Matlab always irritate me since I screw up the scaling and then get confused about why my program doesn't work correctly. Inevitably I always trace the root of the problem back to the fft scaling. Part of the problem resides in the fact that although taking a simple Fourier transform and then its inverse is simplistic and will always give the correct results there are these issues. If the quantity being transformed is located in the center of the window (as ours always is) fftshifting is always required. The fftshifting issue is simple, however, things get tricky since basic NLSE code will work if one ignores the fft scaling so when one modifies the code and then it doesn't work you are left scratching your head: "It worked before! Why not now?" The reason is that the scaling in the fourier domain must be correct if you are going to do autocorrelation or solve the Generalized NLSE for example anything where your are multiplying multiple things in the Fourier domain.

Lets start out by reviewing the fft/ifft definitions in matlab and then work out an example to verify our approach..

Matlab Definitions

Matlab defines its **fft** using the following:

$$\tilde{X}(k) = \sum_{n=1}^{N} x(n) e^{-j2\pi(k-1)(n-1)/N}, 1 <= k <= N \tag{1}$$

This corresponds to the analytic result:

$$\tilde{X}(w) = \int_{-\infty}^{\infty} x(t) e^{-iwt} dt \tag{2}$$

We see that this actually corresponds to our inverse FT although it does not contain the $1/2\pi$ scaling

Matlab defines its **ifft** as:

$$x(n) = \frac{1}{N} \sum_{k=1}^{N} \tilde{X}(k) e^{j2\pi(k-1)(n-1)/N}, 1 <= n <= N \tag{3}$$

This of course corresponds to our Fourier transform:

$$x(t) = \int_{-\infty}^{\infty} \tilde{X}(w) e^{iwt} dw \tag{4}$$

Lets look at the spectral domain of $|\phi|^2\phi$

We start out by defining some time vector T and a width T0 we then define a Gaussian field so that we can check our code:

$$\phi(t) = A0 e^{(-T.^2)/(2 \cdot T0^2)} \tag{5}$$

To get this field in the frequency domain we will use the ifft as the Fourier Transform!

$$\phi(\Omega) = fftshift(ifft[fftshift[\phi(t)]]) \cdot scale \tag{6}$$

For the Gaussian we are looking at:

$$\phi(\Omega) = \sqrt{2\pi} \cdot T0 \cdot exp\left(\frac{-\Omega^2 T0^2}{2}\right) \tag{7}$$

where scale is defined as scale = dT·N

As noted above simply transforming and then back transforming always works but to verify that our code is correct we check against the nonlinear term in the NLSE as noted above

$$old(\Omega) = \phi(\Omega); \tag{8}$$
$$old(t) = fftshift(fft[fftshift[old(\Omega)]])/scale; \tag{9}$$
$$I = abs(old(t)).^2; \tag{10}$$

So we did just go in a circle: We defined our field in the time domain then transformed to the frequency domain. We then transformed back Hence we are just left with what we started with: (but we are doing this to check our code one step at a time)

$$\phi(t) = A0e^{-T^2/(2T0^2)} \tag{11}$$

Now define the nonlinear term:

$$NL(t) = old(t). * I; \tag{12}$$

Analytically this just gives us:

$$NL(t) = A0^3 e^{-3T^2/(2T0^2)} \tag{13}$$

in the time domain

Now to get this in the frequency domain we just perform:

$$\psi(\Omega) = fftshift(ifft[fftshift[NL(t)]]) \cdot scale; \tag{14}$$

Which analytically corresponds to:

$$\phi(\Omega) = A0^3 \sqrt{\frac{2\pi}{3}} T0 e^{-w^2 T0^2/6} \tag{15}$$

NLSE using RKF

The NLSE may be solved in the frequency domain simply by integrating:

$$\frac{\partial \tilde{S}(\Omega, z)}{\partial z} = \frac{i\beta_2}{2}\Omega^2 \tilde{S}(\Omega, z) + i\gamma F\left[|\tilde{S}(t, z)|^2 \tilde{S}(t, z)\right] \tag{16}$$

Using the Runge-Kutta-Fehlberg Method of fifth order:

Looking specifically at higher order soliton propagation we get:

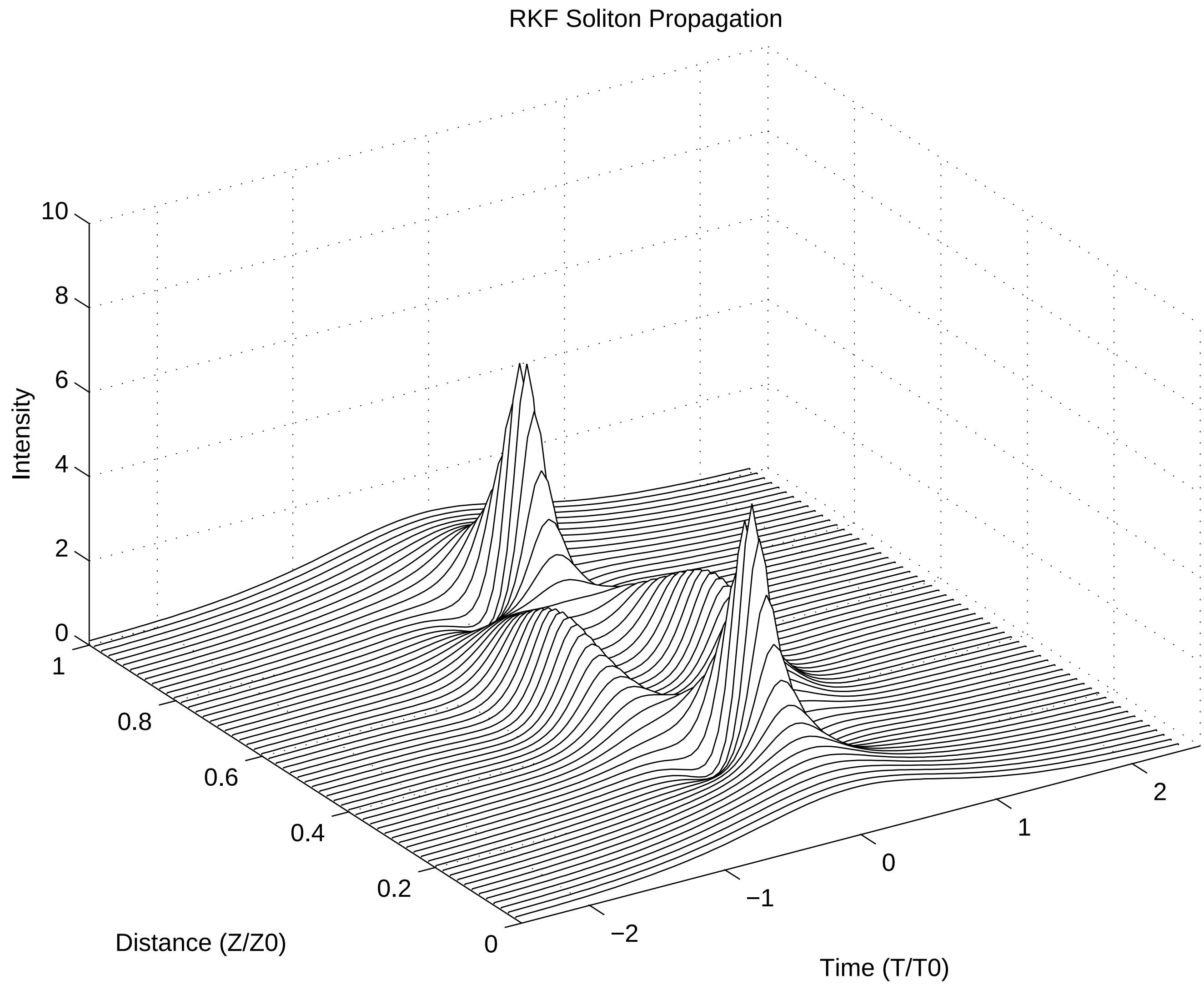

Figure 1: Propagation of a third order soliton over one soliton period using RKF.

Well this is easy to get with the split step so lets look at the next two:

And finally some serious computation to get a nice view of N = 5..

Refrences for this kind of thing: P. L. Francois, "Nonlinear propagation of ultrashort pulses in optical fibers: total field formulation in the frequency domain" JOSAB 276 (1991).

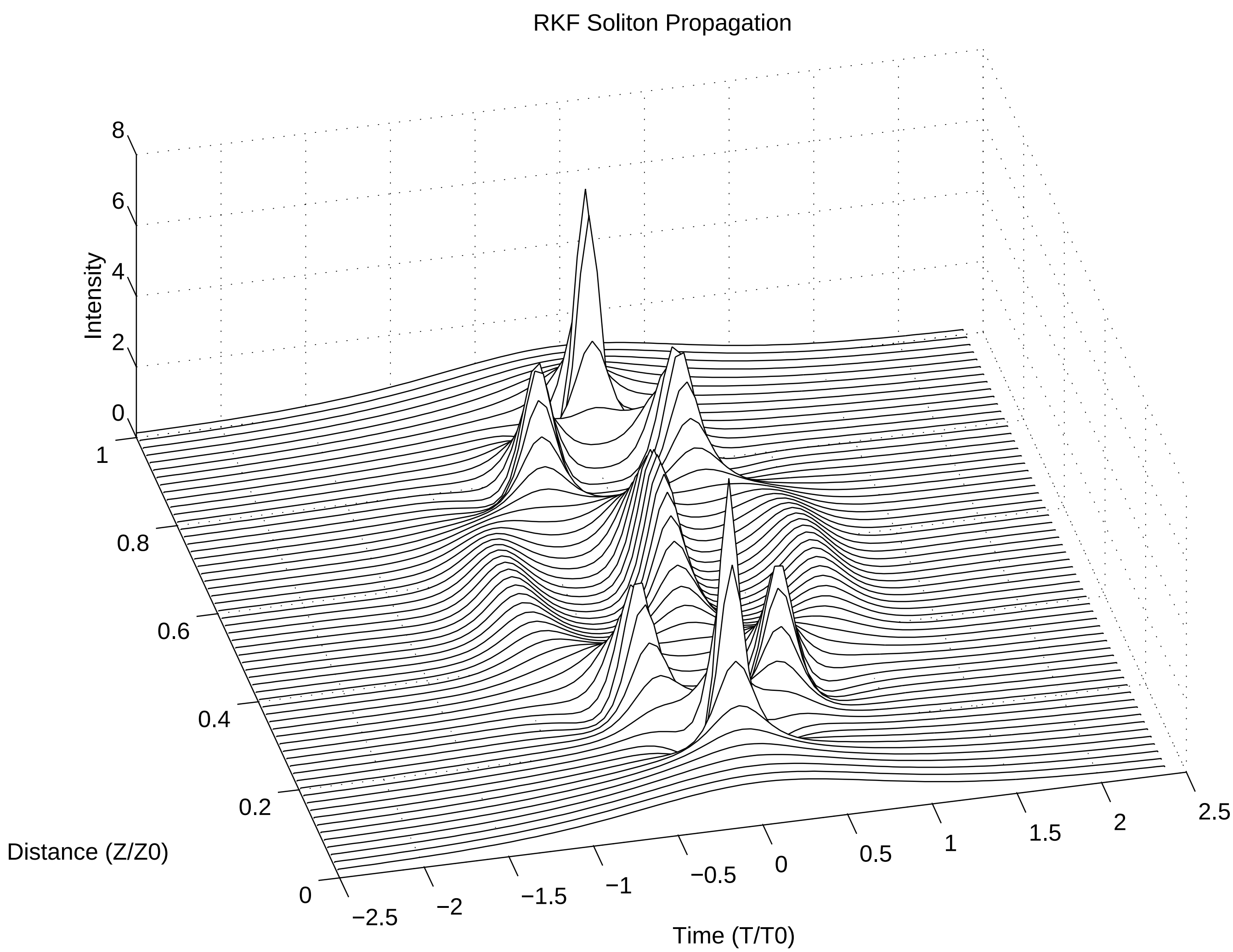

Figure 2: Propagation of a fourth order soliton over one soliton period using RKF.

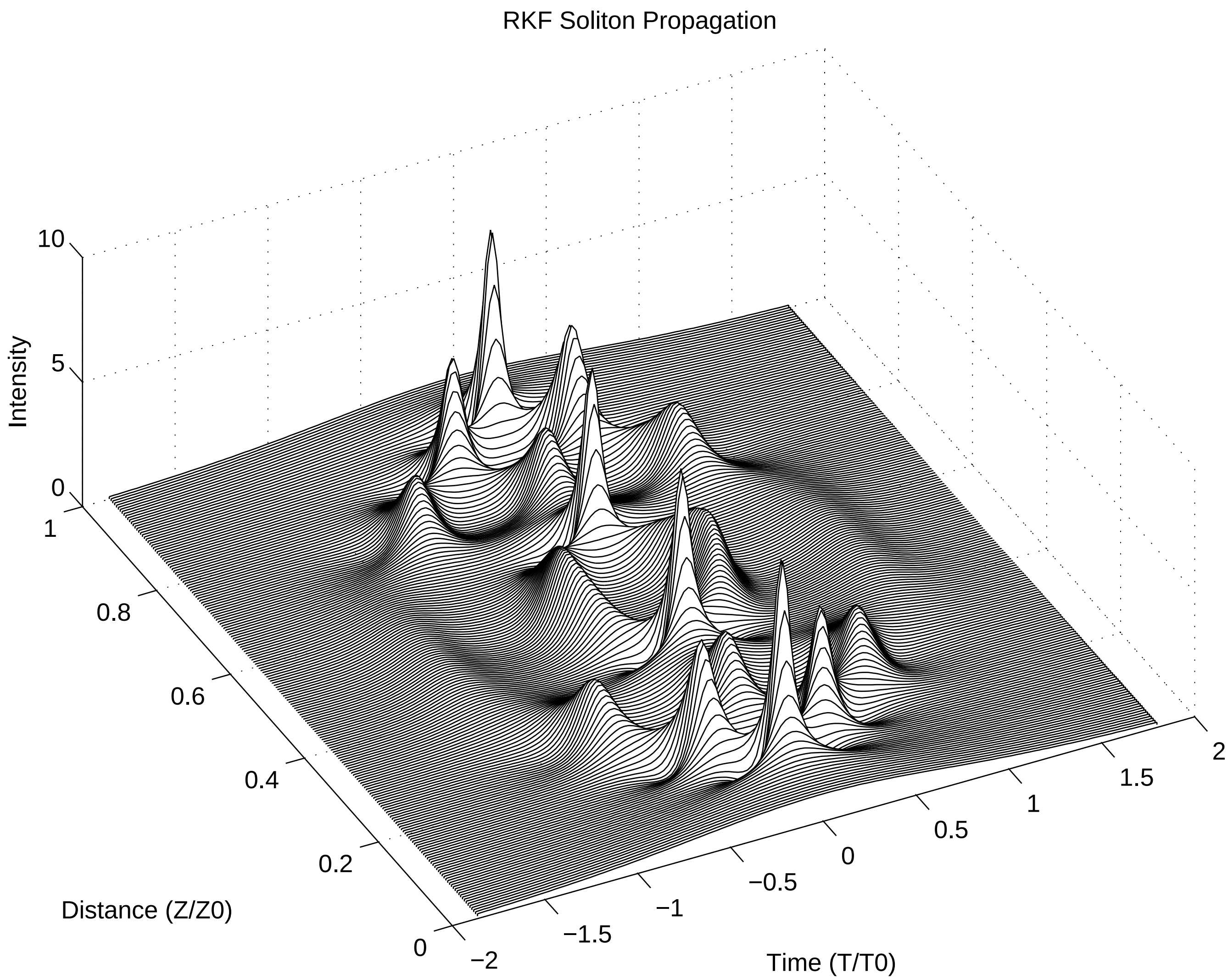

Figure 3: Propagation of a fifth order soliton over one soliton period using RKF.