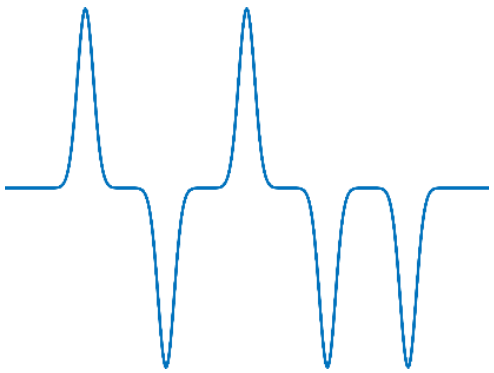
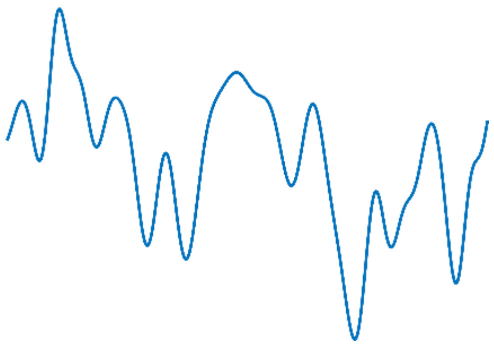



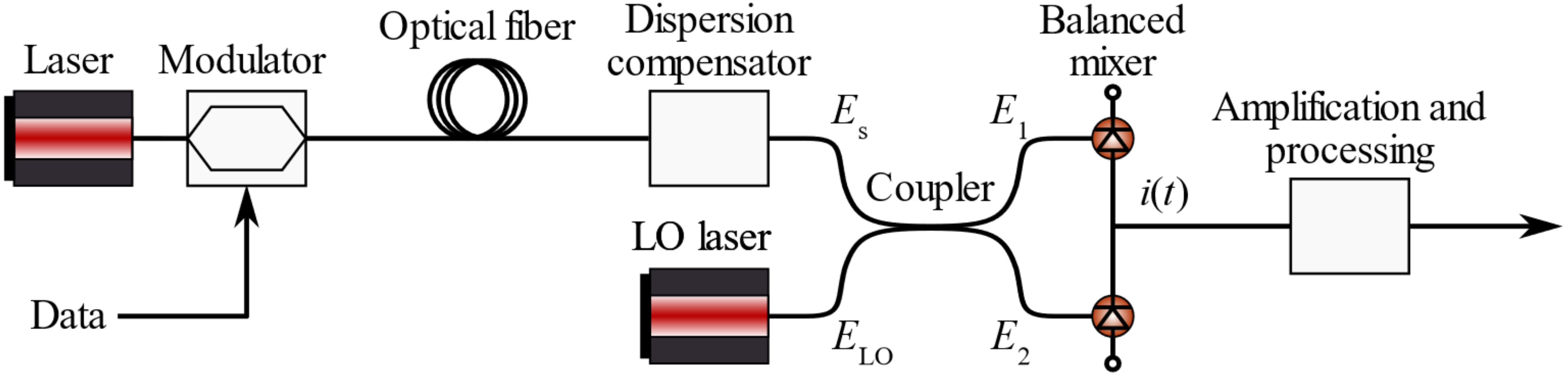
Solution of the nonlinear Schrödinger equation using cuFFT

Ilya Kuk

Instructor:
Sergey Rykovanov
TAs:
Rishat Zagidullin and Egor Konyagin

HPC PROJECT 2021

Coherent communications

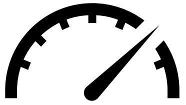
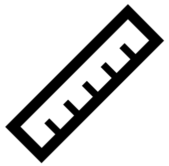


Fiber chromatic dispersion → pulse broadening

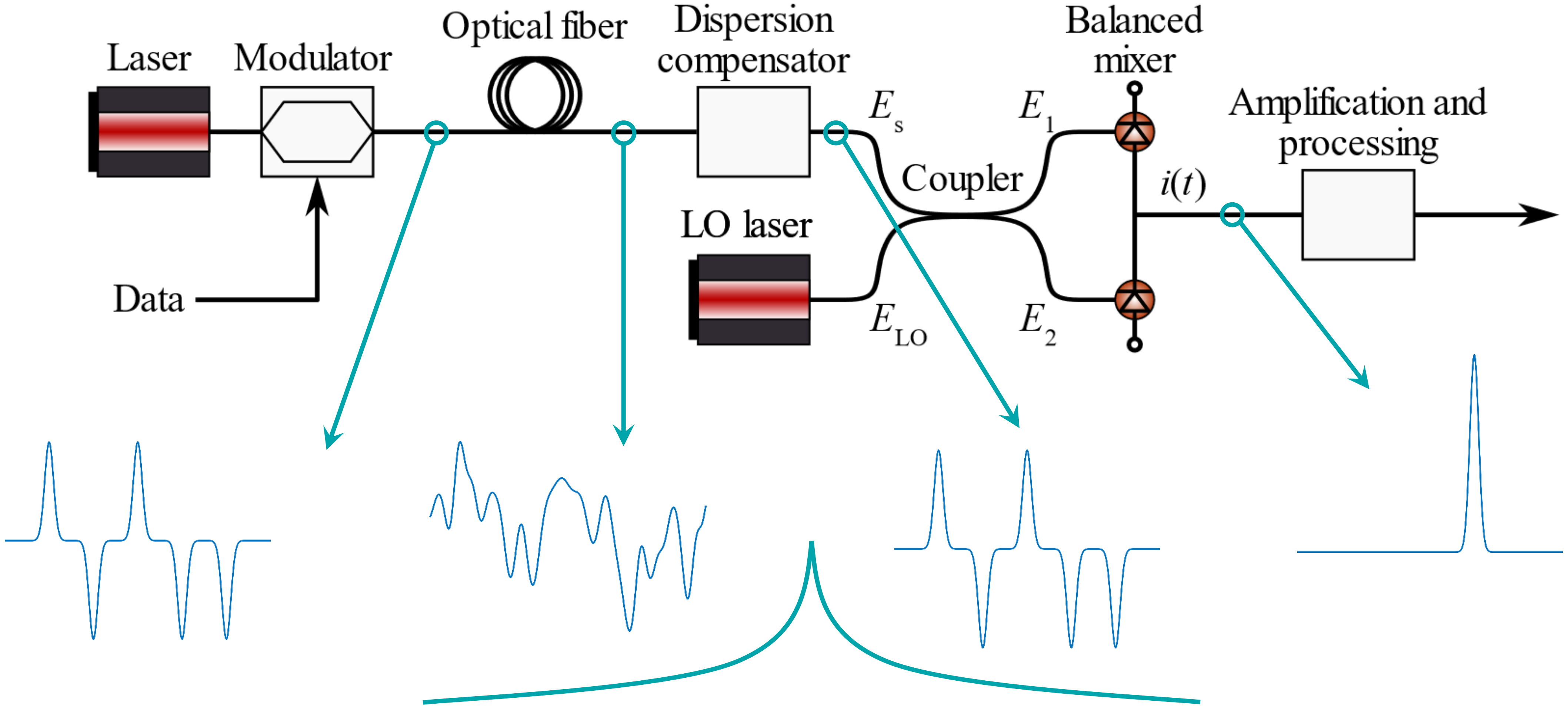
Kerr nonlinearity ($n = n_0 + \alpha|E|^2$) → phase distortion ⇒ errors after decoding.

Nonlinearity is limiting system performance:

higher bit-rate or longer transmission distance

1. **higher bit-rate**: $\mathcal{E} \approx |E|^2\tau_0 \approx |E|^2/BR \geq \mathcal{E}_{cr} \Longrightarrow |E|^2 \geq \mathcal{E}_{cr}BR$ 
2. **Length of nonlinearity**: $z_{nl} \approx (\alpha|E|^2)^{-1}$.
   Nonlinearity becoming noticeable when $L \sim z_{nl}$ 

Waveform change during propagation



Digital back propagation

Modeling equation

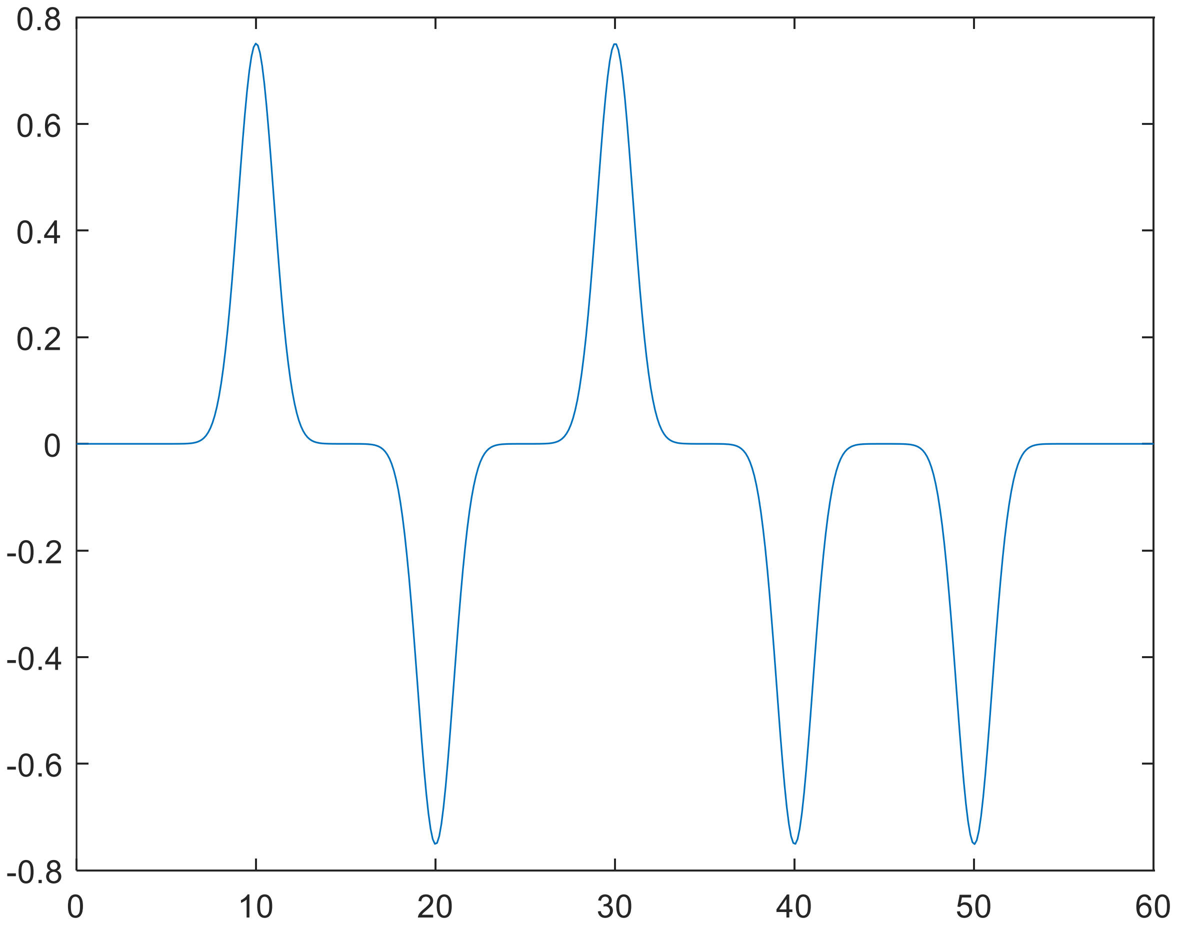
Dimensionless NLS:

$$iE_z + \frac{1}{2}E_{tt} + \varepsilon |E|^2 E = 0 \qquad (1)$$

Bit-sequence launched at the front end of the system is represented by periodic train of **gaussian pulses** with Differential Phase Shift Keying (DPSK)

$$E(t,0) = \sum_{k=1}^{N} a_k \pi^{-1/4} \exp\left[-\frac{1}{2}\left(t - kT^2\right)\right] \qquad (2)$$

$$\begin{aligned} a_k &= 1 \quad \text{with probability} \quad p_1 = 1/2 \\ a_k &= -1 \quad \text{with probability} \quad p_2 = 1/2 \end{aligned} \qquad (3)$$



Split-step method

$$E_z = i\frac{1}{2}E_{tt} + i\varepsilon|E|^2 E = [\hat{D} + \hat{N}]E \tag{4}$$

The equation can be split into a linear part,

$$E_z = i\frac{1}{2}E_{tt} = \hat{D}E \tag{5}$$

and a nonlinear part,

$$E_z = i\varepsilon|E|^2 E = \hat{N}E \tag{6}$$

- Half dispersion step

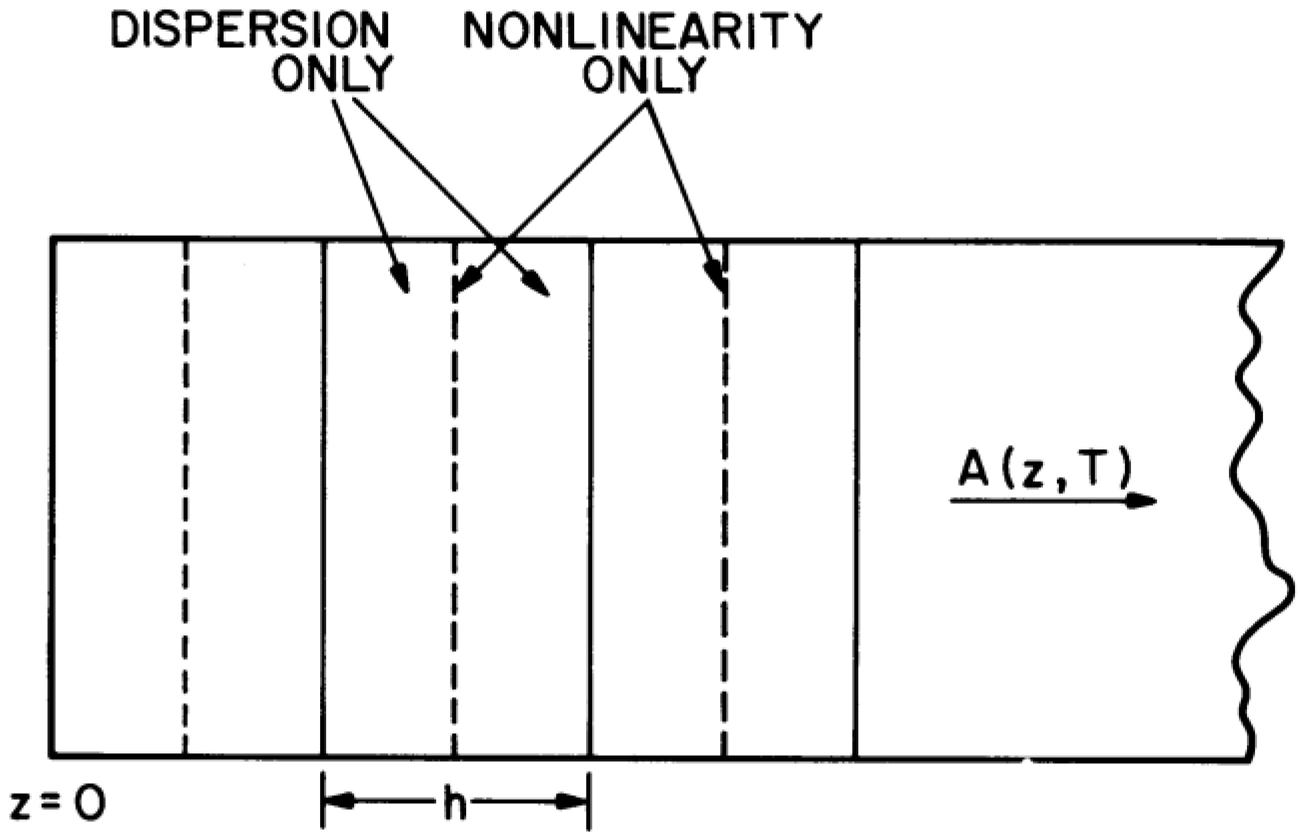
$$E(t, z + \frac{dz}{2}) = F^{-1}(F[E(t,z)] \cdot \exp[-i\frac{1}{2}\frac{dz}{2}w^2]), \tag{7}$$

where F and F^{-1} denotes forward and the reverse Fourier transform respectively.

- Nonlinear step

$$E(t, z + dz) = E(t,z) \cdot \exp[i\varepsilon dz|E|^2] \tag{8}$$

Split-step method



cuFFT some features

Complex type of variables and plan for Fourier transform

```
cufftDoubleComplex *d_LP, *d_u;

cufftHandle plan;
cufftPlan1d(&plan, dim_t, CUFFT_Z2Z, 1);
```

Forward and Inverse Fourier transform

```
cufftExecZ2Z(plan, input, output, CUFFT_FORWARD);
cufftExecZ2Z(plan, input, output, CUFFT_INVERSE);
```

Main split-step loop

```
for (int i = 0; i < int(z_end/z_step); i++)
{
    half_lin(d_u, d_LP, plan, grid, block, dim_t);
    nonlin<<<grid, block>>>(d_u, nonlinearity, z_step, dim_t);
    half_lin(d_u, d_LP, plan, grid, block, dim_t);
}
```

Results

https://github.com/CosmosRedshift7/NLSE-CUDA