8/24/2018 Probably solved the bug of linear mode coupling when “sim.M=1”.

**Problem:** For soliton\_propagation.m case, the results are different when sim.M is and isn’t 1.

**Fix:** Add these lines in “RQ\_matrices = create\_rand\_matrices\_for\_RQ(sim,num\_modes,rand\_func,num\_RQ,std\_Q)” in “func\_of\_linear\_mode\_coupling.m” function.

std\_Q = diag(std\_Q);

idx = zeros(1,num\_modes);

previous\_mg = cumsum(sim.mode\_groups);

previous\_mg = [0 previous\_mg(1:end-1)];

for mg = 1:length(std\_Q)

idx(previous\_mg(mg) + (1:sim.mode\_groups(mg))) = ones(1,sim.mode\_groups(mg))\*mg;

end

RQ\_matrices = rand\_func.hermitian(sim.mode\_groups,num\_RQ).\*std\_Q(idx);

Forgot to multiply the matrix with the coupling strength (with only red texts above) making it really strong in this case. The coupling length becomes approximately 5 meters here. Therefore running the simulation with deltaZ=5 isn’t a good option. I run it with deltaZ=1, M=1 and 20, give the same results.

11/28/2018 Checked the code under single-mode and multimode and compared them all with the original GMMNLSE code in WiseGroup Dropbox folder. Please look at “polarization test” folder under “Examples”. They were also run under the original code. All the result shows that they have even the same numerical values. After these checks, the code I have can be trusted for the cases of single-mode and multimode, with or without isotropic or anisotropic Raman models, under whatever polarization basis chosen, with or without gain model and whatever gain model is used.

The only two things I have nothing to compare with are “linear mode coupling” and “gain rate equation” codes.

11/28/2018 From the check above, a bug in “GMMNLSE\_MPA\_step\_newgain.m” was found.. I accidentally deleted one line of code.

Ra = dt\*fft(haw.\*ifft(Ra));

Put it back. I have no idea how long this bug has existed but fortunately I haven’t needed to use new gain model besides testing gain rate equation model since now.

11/28/2018 Improvement of the code.

**Problem:** When testing gain rate equation model, I found out that multiplying sim.fr and (1-sim.fr) inside split-step or MPA process took some amount of time by looking at “profile on -> profile viewer” in Matlab.

**Fix:** I moved this multiplication to GMMNLSE\_propagate.m incorporating sim.fr with nonlinear coefficients, , , and with the following lines. Besides, some modifications of all the split-step and MPA codes are required but not listed here. Under scalar fields, is calculated from by multiplying it with a SK\_factor. However, since is incorporated with sim.fr, needs to be divided by sim.fr to recover the value. But there’s a problem here: previously if we don’t want Raman term, sim.fr is set to zero. This will cause the division of to fail, so I have to write some codes to calculate term rather than being dependent on .

% Incoporate (1-sim.fr) and sim.fr into SRa,SRb,SK, or kappaK,kappaR1,kappaR2

if sim.lmc\_model ~= 3

if isempty(SK\_info) % scalar fields

if sim.Raman\_model ~= 0

sim.SK\_factor = (1-sim.fr)/sim.fr\*sim.SK\_factor; % For scalar fields, SK is calculated from SRa. Because SRa will be multiplied by sim.fr below and SK also needs to be multiplied by (1-sim.fr), a factor of (1-sim.fr)/sim.fr is included here.

end

else % polarized fields

SK\_info.SK = (1-sim.fr)\*SK\_info.SK;

end

if sim.Raman\_model ~= 0

SRa\_info.SRa = sim.fr\*SRa\_info.SRa;

end

if ~isempty(SRb\_info)

SRb\_info.SRb = sim.fr\*SRb\_info.SRb;

end

else % Manakov

kappaK = (1-sim.fr)\*kappaK;

kappaR1 = sim.fr\*kappaR1;

kappaR2 = sim.fr\*kappaR2;

end

11/28/2018 Rerun the soliton collision in Mecozzi’s paper.

**Problem:** Inconsistent result under weaker random mode coupling.

**Comments:** For strong random mode coupling, it matches quite well with their result; whereas for weaker random mode coupling, it’s different. I think their simulations are run with “strong polarization-mode coupling but weaker spatial-mode coupling”. Since my code can only deal with one correlation length. I ran it with a short correlation length along with a short coupling length between polarization modes but varying coupling lengths between spatial modes. But this gives a different result. I’m not sure if my code is incorrect.