Example Calculation

This document will take the reader step-by-step through an example of a spatial phase calculation for Type-I entangled photon source simulation in Matlab. The top level function is PHASEMAP_AND_RHO.m. The inputs to this function are the parameters of the pump beam, down-conversion crystals, compensation crystals, collection irises and filters. See comments in PHASEMAP_AND_RHO.m for more details on the calculation procedure. Use this code first to find spatial decoherence / spatial compensation results and then use the resulting density matrix in the spectral-temporal code to find joint effects. For this program always input the pump beam as a monochromatic beam,i.e, input the central wavelength and a negligible bandwidth (regardless of the actual pump bandwidth – the spectral effects are calculated separately)

The optical table is in the yz-plane and all crystal surfaces are in the xy-plane. In the simulation, the beam propagates in the z-direction. See setup_diagram.pdf for a schematic of the setup used in this example.

Setting the parameters of the pump beam

```
>> pump = set_pump([cos(pi/4); sin(pi/4); 0], 1000, 1e9, 0.405,
0.000001)

pump =

                D: [3x1 double]
                w: 1000
                R: 1.0000e+009
    lambda_center: 0.4050
     lambda_width: 1.0000e-006
```

- D is the polarization vector of the pump. In this case it is 45 degree polarized.
- w is the radius of the pump at the DC crystals, in microns
- R is the radius of curvature of the pump at the DC crystals in mm. In this case, the crystals are at the waist of the beam so the radius of curvature is essentially infinite. If a 1-m focusing lens was placed just before the crystals, the radius of curvature would be 1000 mm.
- lambda_center is the center wavelength of the pump, in microns
- lambda_width is intentionally entered as a very small number and does not represent the actual pump bandwidth. Enter it as 1E-6 always – since this program only calculates the spatial effects and spectral-temporal effects are calculated separately.

Set the properties of the down-conversion crystals:
Theta and phi are the crystal cut angles specified when the crystal was cut. Gamma is the crystal's angle of rotation about the z-axis. For gamma = 0 the optic axis (for uniaxial crystal) is in the xz-plane. For gamma = 90 deg, the optic axis is in the yz-plane. Theta

and phi are what goes into the crystal parameter specifications and determine the effective nonlinearity. This differs depending on different textbooks and manufacturers. E.g., Newlight photonics defines θ and ϕ as dooe= d31 sinθ +(d11 cos3φ - d22 sin3φ) cosθ, where dooe is the effective nonlinearity and d31 is the corresponding nonlinear coefficient.

```
Crystal 1:
Type = BiBO
Thickness = 600 um
Theta = 151.7 deg
Phi = 90 deg
Gamma = 0

Crystal 2:
Type = BiBO
Thickness = 600 um
Theta = 151.7 deg
Phi = 90 deg
Gamma = 90 deg
```

Set these parameters using the set_crystal_param.m function:

```
>> dc_1 = set_crystal_param('BiBO', 600, 151.7*pi/180, pi/2, 0)

dc_1 =

         crystal_type: 'BiBO'
    crystal_thickness: 600
                theta: 2.6477
                  phi: 1.5708
                gamma: 0

>> dc_2 = set_crystal_param('BiBO', 600, 151.7*pi/180, pi/2, pi/2)

dc_2 =

         crystal_type: 'BiBO'
    crystal_thickness: 600
                theta: 2.6477
                  phi: 1.5708
                gamma: 1.5708
```

Set the parameters of the compensation crystals in the same way as the down-conversion crystals. As shown in setup_diagram.pdf the two compensation crystals are placed in the path of the signal and idler photons the optical axes in the plane of the table. The compensators used in this example are 245-micron thick BBO crystals, and were *not* designed to completely compensate the phase in this source. Crystal type 'BBO1' refers to a particular set of Sellmeier coefficients that were specified by the manufacturer (Newlight Photonics). 'BBO2' refers to coefficients from the Handbook of Nonlinear Crystals .

```
>> comp_s = set_crystal_param('BBO1', 245, 33.9*pi/180, 0, pi/2)
```

```
comp_s =

          crystal_type: 'BBO1'
    crystal_thickness: 245
                 theta: 0.5917
                   phi: 0
                 gamma: 1.5708

>> comp_i = set_crystal_param('BBO1', 245, -33.9*pi/180, 0, pi/2)

comp_i =

          crystal_type: 'BBO1'
    crystal_thickness: 245
                 theta: -0.5917
                   phi: 0
                 gamma: 1.5708
```

Set the parameters of the pre-compensation crystal, which creates a delay between the two polarization components of the pump.

```
>> pre_comp = set_crystal_param('Quar', 15000, pi/2, 0, pi/2)

pre_comp =

          crystal_type: 'Quar'
    crystal_thickness: 15000
                 theta: 1.5708
                   phi: 0
                 gamma: 1.5708
```

Set collection interference filter parameters:
Signal filter center wavelength = 773 nm
Signal filter bandwidth = 10 nm
Idler filter center wavelength = 851 nm
Idler filter bandwidth = 20 nm
Number of wavelengths used to model the signal bandwidth=3
Number of wavelengths used to model the idler bandwidth=3

The filter parameters are set using:

```
>> filters = set_filters(0.773, 0.01, 0.851, 0.02, 3, 3)

filters =

      lambda_s_array: [0.7680 0.7730 0.7780]
      lambda_i_array: [0.8410 0.8509 0.8610]
    lambda_s_center: 0.7730
            width_s: 0.0100
    lambda_i_center: 0.8510
            width_i: 0.0200
               N_sf: 3
               N_if: 3
```

Note all filter units are in microns. The N_sf and N_if variables are the number of wavelengths within the signal bandwidth and idler bandwidth, respectively, that the program will calculate. So in this case, the pure state calculation will be carried out for the idler wavelengths:

```
lambda_s_array =

    0.7680   0.7730   0.7780
```

(Notice that these are evenly distributed in frequency space, not wavelength space)

Set the Collection iris parameters:

The down-conversion (DC) crystals were cut to be phasematched when pumped at 405 nm. The cone angles were determined to be 3.07 deg for the signal photons at 773 nm and 3.39 deg for the idler photons at 851 nm using the NIST Noncollinear Phasematching Software (http://physics.nist.gov/Divisions/Div844/facilities/cprad/PMProgram.html). The irises are 1200 mm from the DC crystals. The signal and idler irises are opened to a 1.5 mm radius. To save calculation time, keep the number of points per iris small (6 in this case). The calculation time scales as the product of all the N's, i.e., N_sf*N_sx*N_sy*N_if* N_ix*N_iy

```
>> irises = set_irises([0;64;1200], 1.5, [0;-71;1200], 1.5, 6, 6,
6, 6)

irises =

    s_x_array: [-1.5000 -0.9000 -0.3000 0.3000 0.9000 1.5000]
    s_y_array: [62.5000 63.1000 63.7000 64.3000 64.9000 65.5000]
    i_x_array: [-1.5000 -0.9000 -0.3000 0.3000 0.9000 1.5000]
    i_y_array: [-72.5000 -71.9000 -71.3000 -70.7000 -70.1000 -
69.5000]
     center_s: [3x1 double]
     radius_s: 1.5000
     center_i: [3x1 double]
     radius_i: 1.5000
         N_sx: 6
         N_sy: 6
         N_ix: 6
         N_iy: 6
```

Also, set comp_on = 1 to turn on the spatial compensators and custom_coeff=0

Finally, run the top level function (the calculation will run for ~ 3 minutes):

```
>> [rho, Tangle, phases, irises, param] = PHASEMAP_AND_RHO(irises,
filters,...
pre_comp, dc_1, dc_2, comp_s, comp_i, pump, comp_on, custom_coeff)

rho =
   0.4997             0.0016 - 0.0022i   0.0112 - 0.0001i  -0.4872
- 0.0174i
   0.0016 + 0.0022i   0.0003            -0.0001 + 0.0001i  -0.0037
- 0.0023i
   0.0112 + 0.0001i  -0.0001 - 0.0001i   0.0005            -0.0089
+ 0.0002i
  -0.4872 + 0.0174i  -0.0037 + 0.0023i  -0.0089 - 0.0002i   0.4995
Tangle =
    0.9498
phases =
       phi_ext_s_dc: [4-D double]
       phi_ext_i_dc: [4-D double]
    phi_pump_1_fast: [4-D double]
    phi_pump_1_slow: [4-D double]
    phi_pump_2_fast: [4-D double]
      phi_pump_pc_1: [4-D double]
      phi_pump_pc_2: [4-D double]
              dc_s1: [6-D double]
              dc_s2: [6-D double]
              dc_i1: [6-D double]
              dc_i2: [6-D double]
                 dc: [6-D double]
        total_phase: [6-D double]
            x_array: [-1.5000 -0.9000 -0.3000 0.3000 0.9000 1.5000]
            y_array: [62.5000 63.1000 63.7000 64.3000 64.9000
65.5000]
         dc_x_slope: 15.0486
         dc_y_slope: -0.0011
       comp_x_slope: -13.6212
       comp_y_slope: 0.0204
      total_x_slope: 1.4274
      total_y_slope: 0.0193
irises =
    s_x_array: [-1.5000 -0.9000 -0.3000 0.3000 0.9000 1.5000]
    s_y_array: [62.5000 63.1000 63.7000 64.3000 64.9000 65.5000]
    i_x_array: [-1.5000 -0.9000 -0.3000 0.3000 0.9000 1.5000]
    i_y_array: [-72.5000 -71.9000 -71.3000 -70.7000 -70.1000 -
69.5000]
     center_s: [3x1 double]
     radius_s: 1.5000
     center_i: [3x1 double]
     radius_i: 1.5000
         N_sx: 6
         N_sy: 6
         N_ix: 6
         N_iy: 6
       dist_s: [4-D double]
       dist_i: [4-D double]
param =
      lambda_pump: [0.4050 0.4050 0.4050]
    lambda_signal: [0.7680 0.7730 0.7780]
     lambda_idler: [3x3 double]
```

```
        iris_hits: 96
         max_hits: 324
```

rho is the density matrix representing the collected two-photon polarization state. The tangle T, is the square of the concurrence (see PHASEMAP_AND_RHO.m for details).

The "slopes" in the above “Phases” list are the slopes in the x and y directions of the phasemaps over the signal iris. The phase is the phase difference between the VV and HH terms of the two-photon state. This phase is different for each position on the iris, leading to a phase-gradient (angle/spatial dimension) or slope. The dc slopes are for maps of the phase acquired only in the DC crystals, and likewise for the comp (compensator) slope. The total slope is the phase map slope due to all the crystals.

NOTE: This program only calculates the effects of birefringence and DOES NOT CALCULATE the effective nonlinearity ⇒ there might not be any downconversion photons even though phasematching says that there might be.