# REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-09-0310

| 1. REPORT DATE (DD-MM-YYYY) | 2. REPORT TYPE | 3. DATES COVERED (From - To) |
|---|---|---|
| 31-08-2009 | Final | Nov 2008 - Jul 2009 |

**4. TITLE AND SUBTITLE**
Reconfigurable Optical Elements Based on Single and Coupled Microdisk Resonators with Quantum Dot Active Media

**5a. CONTRACT NUMBER**
FA9550-09-C-0047

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**

**6. AUTHOR(S)**
Menon, Vinod

**5d. PROJECT NUMBER**

**5e. TASK NUMBER**

**5f. WORK UNIT NUMBER**
0001AC

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
HYBRID PHOTONICS, LLC
40 IU WILLETS RD
ALBERTSON NY 11507-1338

**8. PERFORMING ORGANIZATION REPORT NUMBER**
HYB0003Z

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
AF OFFICE OF SCIENTIFIC RESEARCH
875 N. RANDOLPH ST. ROOM 3112
ARLINGTON VA 22203

**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFOSR

**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution unlimited

**20091116191**

**13. SUPPLEMENTARY NOTES**

**14. ABSTRACT**
We report the results of Phase I efforts to realize reconfigurable photonic systems relying on optical bistability in microring and microdisk structures. These reconfigurable systems that can function as optical memory elements are expected to have applications in all-optical signal processors. Hybrid Photonics investigated optical bistable devices based on novel physical principles, which distinguish our efforts from all previously suggested approaches. Two configurations were studied: (i) two active waveguides side coupled to a ring resonator, and (ii) two coupled active microdisk resonators. In both configurations CdSe core-shell quantum dots were used as active gain media. Results of our theoretical efforts at understanding the underlying physical mechanism for the bistability and the modeling efforts to design the bistable device are presented. We also describe the experimental work on the two device configurations and the results obtained. Bothe theoretical and experimental work performed indicate the coupled microdisk geometry to be the more practical approach for achieving bistable lasing – a necessary condition for demonstrating an all-optical flip-flop. Finally, we summarize our findings and make recommendations for future work based on our phase I results.

**15. SUBJECT TERMS**
STTR Report, Reconfigurable Optical Elements, Hysteresis, Microdisk Resonators, Quantum Dots, Lasers, Bistability

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Vinod Menon |
| U | U | U | UU | | 19b. TELEPHONE NUMBER (Include area code) (609) 575-1752 |

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18
Adobe Professional 7.0

# RECONFIGURABLE OPTICAL ELEMENTS BASED ON SINGLE AND COUPLED MICRODISK RESONATORS WITH QUANTUM DOT ACTIVE MEDIA

*Principal Investigator: Dr. Vinod Menon*
*Hybrid Photonics, LLC*
*40 I.U. Willets Rd., Albertson, NY 11507*
*Phone: 516-747-9417, E-mail: hphotonics@physics.qc.edu*

## FINAL PERFORMANCE REPORT

## TABLE OF CONTENTS

## LIST OF FIGURES

## GLOSSARY

CUNY - The City University of New York

MMI - multi-mode interference

PICs - photonic integrated circuits

QDs - quantum dots

## 1.0    INTRODUCTION

Chip-scale multifunctional photonic integrated circuits (PICs) are critical for realizing high performance components required in modern optical data transmission links and telecommunication systems. Such PICs are expected to play a crucial role in next generation smart and adaptive sensors that would require the development of high performance optical signal processors and data routing technologies that will facilitate efficient and fast transfer of information, on-demand connection and ultrafast data processing. They are also essential for reducing the production cost of these systems. An important sub-component in realizing practical all-optical signal processor is an optical memory (buffer) element. Most suggestions for all-optical memories and flip-flops are based on the phenomena of optical bistability, when a nonlinear optical system can operate at two different output intensities for the same input intensity. In most cases the nonlinearity in semiconductor based structures is due to an intensity dependent change of refractive index produced by photo-generated electron-hall pairs. Switching times of these devices is limited by recombination times of the carriers, which is of the order of nanoseconds.

In the Phase 1 efforts undertaken at Hybrid Photonics in cooperation with Queens College of CUNY we investigated optical bistable devices based on novel physical principles, which distinguish our efforts from all previously suggested approaches. Two configurations were studied: (i) two active waveguides side coupled to a ring resonator, and (ii) two coupled active microdisk resonators. In both configurations CdSe core-shell quantum dots (QDs) were used as active gain media.

In Section 2, we discuss our theoretical efforts at understanding the underlying physical mechanism for the bistability and the modeling efforts to design the bistable device. Following this in Section 3, we describe the experimental work on the two device configurations and the results. Finally, in Section 4, we summarize our findings and discuss the future plans.

## 2.0    THEORETICAL MODEL OF BISTABILITY IN THE COUPLED WAVEGUIDE-RING GEOMETRY

The first step in designing an efficient device demonstrating the bistable behavior observed in preliminary experiments is to achieve understanding of underlying physical causes

of the observed behavior and to develop a computer model allowing us to relate technical characteristics of the device to its structural parameters. This step is divided in two subtasks: 1) to develop a theoretical model of the modes of the waveguides-resonator system, and 2) to develop a dynamic model of lasing in this structure.

## 2.1. Modes and resonant frequencies of the system

The role of the laser "cavity" in our system is played by two waveguides coupled to each other through the ring resonator. The feedback is provided by reflection of light from the cleaved edges of the waveguides and redirection of the reflected light to the second waveguide by coupling to the ring resonator. We present the field in each waveguide as

$$E_\alpha(z) = A_\alpha \left[ r_\alpha e^{iq_\alpha z} + e^{-iq_\alpha z} \right], z < L_\alpha \tag{1}$$

where $A_\alpha$ and $q_\alpha$ are stationary amplitudes and propagation constants of the field in the waveguides ($\alpha = 1, 2$), $z$ is the coordinate along each of the waveguide with zero chosen at the cleaved edges, and $L_\alpha$ are distances from the edges of the respective waveguides to the coupling region between the waveguides and the ring (since the size of the coupler is much smaller than the wavelength, we can assume that the coupling occurs at a single point), and $r_\alpha$ are reflection coefficients from the cleaved edges of the waveguides. Using standard model of the coupler we derived the relation between the amplitudes of the field in the 1$^{st}$ and 2$^{nd}$ waveguides in the form:

$$A_1 = \eta(\Omega) r_2 A_2 e^{i(q_1 L_1 + q_2 L_2)}$$
$$A_2 = \tilde{\eta}(\Omega) r_1 A_1 e^{i(q_1 L_1 + q_2 L_2)} \tag{2}$$

where

$$\eta(\Omega) = \frac{|\kappa|^2 \sqrt{1-\xi^2} e^{i\Phi(\Omega)/2 - \gamma_r L_r/4}}{1 - |t|^2 (1-\xi^2) e^{i\Phi(\Omega) - \gamma_r L_r/2}}$$

$$\tilde{\eta}(\Omega) = \frac{|\kappa|^2 \sqrt{1-\xi^2} e^{-i\Phi(\Omega)/2 - \gamma_r L_r/4}}{1 - |t|^2 (1-\xi^2) e^{-i\Phi(\Omega) - \gamma_r L_r/2}} \tag{3}$$

In these expressions we introduced the phase change of the field inside the ring

$$\Phi = m_2 \frac{\Omega n_r L_r}{c} = 2\pi m_2 \frac{\Omega}{\Omega_r}; \Omega_r = \frac{2\pi c}{n_r L_r} \tag{4}$$

where $n_r$ and $L_r$ are refractive index and the circumference of the ring resonator, $c$ is the speed of light, and $m_2$ is an arbitrary integer. We also introduced ring's absorption coefficient, $\gamma_r$, parameters of the coupler $t, \kappa$, which obey energy conservation condition $|t|^2 + |\kappa|^2 = 1$, and losses at the coupler $\xi$. Combining Equation (2) and (3) we obtain for real and imaginary parts of the propagation constants $q_{1,2} = q'_{1,2} + iq'_{1,2}$:

$$q'_1 L_1 + q'_2 L_2 = \pi m_1$$
$$q''_1 L_1 + q''_2 L_2 = -\frac{1}{2} \ln S(\Omega) \tag{5}$$

where

$$S = \frac{|\kappa|^4 (1-\xi^2) r_1 r_2 e^{-\frac{\gamma_r L_r}{2}}}{1 + R^2 |t|^4 e^{-\alpha L_r} - 2(1-\xi^2)|t|^2 e^{-\gamma_r L_r} \cos\left(2\pi m_2 \frac{\Omega}{\Omega_r}\right)} \tag{6}$$

Equation (5) shows that the resonance frequencies of the effective "cavity" in this configuration are determined by properties of the waveguides, while the ring resonator affects only the effective losses of the system. The normalized modes of the cavity are defined as

$$U_\alpha(z) = \frac{r_\alpha e^{iq_\alpha z} + e^{-iq_\alpha z}}{\int_0^{L_\alpha} \left| r_\alpha e^{iq_\alpha z} + e^{-iq_\alpha z} \right|^2 dz} \tag{7}$$

## 2.2. Dynamic equations

With normal modes and resonant frequencies determined we can derive dynamic equations for field amplitudes in each of the waveguides. We present field in each waveguide as

$$E_\alpha(t) = \frac{1}{2}\left[ A_\alpha(t) e^{-i\Omega t + i\psi(t)} U_\alpha(z) + c.c. \right] \tag{8}$$

where $\alpha = 1, 2$, $A_\alpha$ and $\psi$ are slow changing amplitude and the phase of the field, $\Omega$ is an unknown lasing frequency, and use standard semiclassical lasing theory [1] to derive dynamic equations for the amplitude and the phase in each waveguide:

$$2\frac{d|A_\alpha|}{dt}+|A_\alpha|v''_\alpha = 2\pi\varsigma^2\Omega g(\Omega)\Delta\rho_0|A_\alpha(t)|\int_0^{L_\alpha}\frac{|U_\alpha(z)|^2}{(1+R_\alpha/R_s)}dz$$

$$\Omega+\dot\psi_\alpha-v'_\alpha = 2\pi\varsigma^2\Omega g(\Omega)\Delta\rho_0\frac{\omega_0-\Omega}{\gamma_\perp}\int_0^{L_\alpha}\frac{|U_\alpha(z)|^2}{(1+R_\alpha/R_s)}dz \qquad (9)$$

where $v'_\alpha$ and $v''_\alpha$ are real and imaginary parts of the resonance frequencies of the lasing cavity, $\varsigma$ is normalized oscillator strength of optical transitions responsible for lasing, $g(\Omega)$ is the gain profile of the active medium centered at the transition atomic frequency $\omega_0$ with effective width $\gamma_\perp$, $\Delta\rho_0$ is unsaturated population inversion, which characterizes the pumping intensity, $R_s$ is saturation parameter, and

$$R_\alpha = \frac{1}{2\hbar}|A_\alpha(t)|^2\varsigma^2 g(\Omega)|U_\alpha(z)|^2 \qquad (10)$$

is responsible for the gain saturation. For the parameters of the structure realized experimentally the lasing is expected to occur at a frequency far above the threshold frequency of the waveguide so that we can assume linear relationship between the propagation constants of the waveguide and the respective frequencies: $v_\alpha \approx c_w q_\alpha$, where $c_w$ is the speed of light in the waveguide. In this case we can combine Equation (9) with Equation (5) to obtain closed system of dynamic equations in the form

$$\dot\psi_1+\dot\psi_2+\Omega-\tilde\varsigma^2\Omega g(\Omega)\frac{\omega_0-\Omega}{2\gamma_\perp}\Delta\rho_0\left[\int_0^{L_\alpha}\frac{|U(z)|^2}{(1+R_u/R_s)}dz+\int_0^{L_\alpha}\frac{|U(z)|^2}{(1+R_d/R_s)}dz\right]=\frac{\pi m_1 c_w}{2L}$$

$$-2\left(\frac{1}{|A_1|}\frac{d|A_1|}{dt}+\frac{1}{|A_2|}\frac{d|A_2|}{dt}\right)+\tilde\varsigma^2\Omega g(\Omega)\Delta\rho_0\left[\int_0^{L_\alpha}\frac{|U(z)|^2}{(1+R_u/R_s)}dz+\int_0^{L_\alpha}\frac{|U(z)|^2}{(1+R_d/R_s)}dz\right] \qquad (11)$$

$$=-\frac{c_w}{2L}\ln S(\Omega)$$

where we assumed that the waveguides are identical. In the stationary regime these equations give an equation for the lasing frequency:

$$\Omega+\frac{c(\omega_0-\Omega)}{4\gamma_\perp L}\ln S(\Omega)=\frac{\pi m_1 c}{2L} \qquad (12)$$

Owning to the oscillating nature of the ring resonance parameter $S$ this equation might admits multiple solutions (see Figure 1 below). We have developed a computer code written in Matlab for solving Equation (12) and determined that for typical parameters of the structure this

equation indeed might have three solutions within the gain width of the atomic transition. Since the gain parameter for different frequencies is different each of the possible frequencies will result in different lasing intensities. Thus, the presence of multiple solutions for the lasing frequency is responsible for the multistable behavior of lasing output in the structure under consideration. In order to determine stability of possible solutions we need to analyze the dynamic behavior of the amplitudes in the vicinity of possible stationary values of the amplitudes. To this end we use the relation between field amplitudes in different waveguides given by Equation (2) in order to derive a dynamic equation for a single intensity, which can be presented in the following form

$$\frac{dI}{dt} = \frac{1}{2}\tilde{I}\left\{\varsigma^2\Omega g(\Omega)\Delta\rho_0 \int_0^L\left[\frac{|U(z)|^2}{1+\beta^2 I\varsigma^2|U(z)|^2 g(\Omega)} + \frac{|U(z)|^2}{1+I\varsigma^2|U(z)|^2 g(\Omega)}\right]dz + \frac{\omega_B}{\pi}\ln S(\Omega)\right\} \quad (13)$$

where $I$ is dimensionless intensity, and $\beta = e^{-\gamma_r L_r/4}$. Frequency is also a dynamic variable whose initial dynamics occurs over very short times of the order of $\gamma_\perp^{-1}$ and can be neglected. In this case the frequency in Equation(13) can be related to the instantaneous value of the intensity as

$$m\omega_B = \Omega\left\{1 - \varsigma^2 g(\Omega)\frac{\omega_0 - \Omega}{2\gamma_\perp}\Delta\rho_0 \int_0^L\left[\frac{|U(z)|^2}{1+\beta^2 I\varsigma^2|U(z)|^2 g(\Omega)} + \frac{|U(z)|^2}{1+I\varsigma^2|U(z)|^2 g(\Omega)}\right]dz\right\} \quad (14)$$

and may evolve to one of the possible stationary values depending on their stability and the initial conditions. To study the stability of possible stationary solutions we used Equations (13) and (14) in the small signal and plot the right hand side of Equation (13) as a function of intensity (Fig. 1). It passes through zero at the stationary solutions and the sign of the slopes of this function in the vicinities of zeroes indicates the stability of the respective solution: negative slope corresponds to stable solutions, and positive to the unstable ones. The black curve in this figure corresponds to a situation when a zero intensity (nonlasing) solution is unstable, and there is a single stable lasing solution with nonzero intensity. With increase of pumping (blue curve) the stability of the solutions changes: zero intensity (non-lasing) solution becomes stable again and coexists with a stable lasing solution. This is a rather unusual situation, when depending on initial conditions the system can either laze or not at the same pumping level. Further increase of pumping changes the situation again. Now the non-lasing solution becomes unstable and two stable lasing solutions emerge. In order to illustrate this situation more clearly we plot in Figure

2 the time dependence of intensity for the same three pumping intensities as in Figure 1, and for small initial intensity, $I_0 = 0.1$.
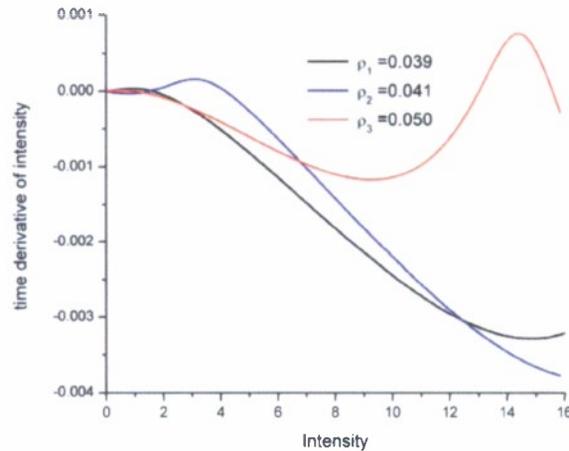


**Figure 1: Time Derivative of Intensity as Function of Intensity. Zeroes Show Stationary Solutions**
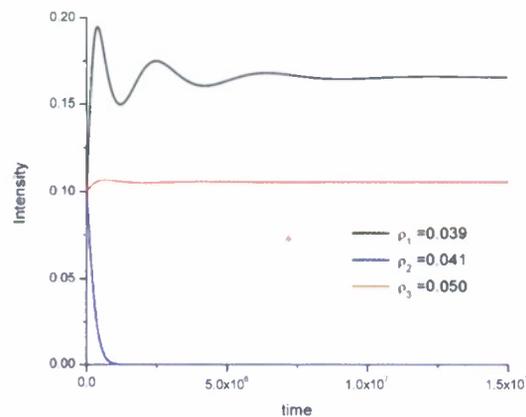


**Figure 2: Time Dependence of Intensity for Small Initial Intensity**

One can see that for small pumping (black curve) the intensity approaches a non-zero value, for the second level of pumping (blue curve) the lasing is quenched for this initial condition, while the further increase of pumping results again in a non-zero stationary intensity. In order to demonstrate the existence of the second stable solution at blue pumping, we plot the

time dependence of intensity for this pumping level, but with different initial condition in Figure 3, which approaches a non-zero solution.
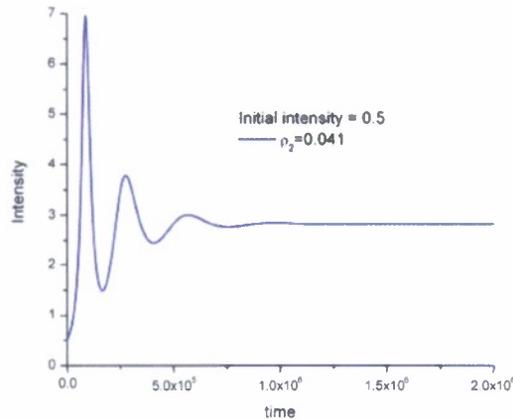


**Figure 3: Time-Dependence of Intensity for the Blue Pumping and Larger Initial Intensity**

Thus we can conclude that the developed model gives a correct physical picture of the observed effect and can serve as a modeling tool for designing structures with desirable bistable and reconfigurable properties. Even though we did not manage to observe these effects with the experimental design studied during the Phase-1 efforts, the obtained theoretical results proof that the general concept of the device based on resonant coupling of the waveguides is correct and can be realized with more carefully prepared structure.

## 2.3. Evanescent coupling and normal modes of coupled disks.

The second thrust of the Phase-I efforts was concerned with observing bistable behavior in the system of coupled microdisks. In order to provide theoretical supports to these efforts and assist in designing of the structure a theoretical model and relevant software for analysis of normal frequencies and modes of single and double-disk structures were developed.

### 2.3.1.  Single disks – resonance frequencies and normal modes

It is well known that microdisk resonators can be with a good accuracy modeled as tow-dimensional structures, in which two different polarizations of the field and can be separated [2].

7

As a result the problem of finding resonance frequencies of these structures is reduced to solving a scalar tow-dimensional wave equation. However, when introducing normal modes of these structures, which are required for modeling their lasing properties one needs to take into account that micro-disk resonators have relatively low Q-factors, which means that their radiative losses must be taken seriously. Analysis carried out in this work was based on so called constant flux modes introduced recently in Ref. [3], which were defined in the following way. Inside the disks the field is described by Bessel function $J_m\left(k_{mn} n_d r\right)$ where $k_{mn} = \omega_{mn}/c$ determines the normal frequency $\omega_{mn}$, while outside of the disk, the field is given by an outgoing Hankel function $H_m\left(kr\right)$, which depends on an external parameter $k = \omega/c$, where spectral parameter $\omega$ does not coincide with the normal frequency. The main advantage of the modes defined this way compared to more traditional quasimodes [4] is that, unlike the latter, they do not diverge at infinity and can, therefore, be directly used to calculate the outgoing intensity of respective lasing structures. Normal frequencies for TE polarized modes in this case are defined by equation:

$$\frac{J_{mn}\left(n_d k_{mn} R\right)}{J'_{mn}\left(k_{mn} n_d R\right)} = \frac{1}{n_d} \frac{k_{mn}}{k} \frac{H_m\left(kR\right)}{H'_m\left(kR\right)} \tag{15}$$

where $n_d$ is the effective refractive index of the disk, These frequencies are complex –valued and depend on the external spectral parameter. The real part of the normal mode is defined as solution of the self-consistent equation $\mathrm{Re}[k_{mn}(k_r)] = k_r$, while its imaginary part is given as $\mathrm{Im}[k_{mn}(k_r)]$. In order to numerically evaluate these quantities we developed computer codes (Appendix A) based on the Mathematica platform using its standard library functions. The choice of Mathematica was due to the fact that it allowed carrying out calculations of Bessel functions with arbitrary precision which was crucial for finding normal frequencies of modes with high values of azimuthal number $m$. Figure 4 below presents the found frequencies and their imaginary parts for modes with azimuthal numbers from 10 to 100 and radial numbers. We also compared the results found for the constant flux modes with those found for quasi-modes. The real and imaginary parts of the latter coincide with those of scattering resonances found as poles of the respective scattering amplitudes.
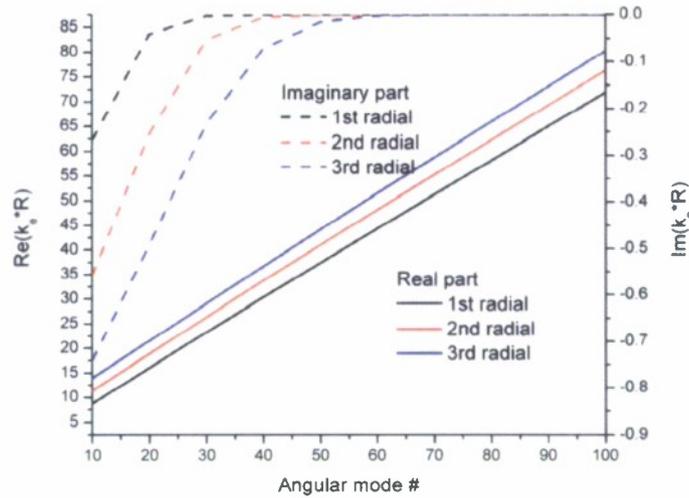
**Figure 4: Real and Imaginary Parts of Normal Frequencies of Constant Flux Modes for Azimuthal Mode Numbers from 10 to 100. Different Colors Correspond to Different Radial Numbers**

The comparison, shown in Figure 5, indicates that while this difference might be significant under some circumstances, it is small enough to be neglected for the purpose of analyzing effects of evanescent coupling on normal modes of multiple disks. Since numerical analysis of the scattering resonances is less numerically expansive, the problem of normal modes of the coupled disks was considered as the scattering problem.
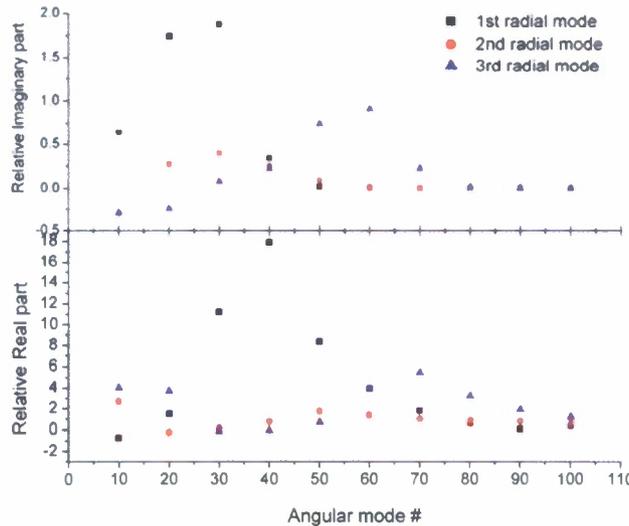


**Figure 5: Relative Differences between Real and Imaginary Parts of Constant Flux Normal Frequencies and Scattering Resonances: It is Seen that for High Order Modes the Difference Becomes Very Small**

### 2.3.2. Normal modes of the double-disk structures

To model a double-disk structure it was found convenient to solve a more general multiple-disk scattering problem. Each disk constituting the structure was characterized by its internal and scattered fields presented in the form of the following modal expansion

$$B_{sc} = \sum_m b_m^{(i)}(\omega) e^{im\varphi_i} H_m \left( n_0 k |\mathbf{r} - \mathbf{r}_i| \right)$$

$$B_{int}^{(i)} = \sum_m d_m^{(i)}(\omega) e^{im\varphi} J_m \left( n_d k |\mathbf{r} - \mathbf{r}_i| \right)$$

(16)

where $i$ enumerates disks characterized by radii $R_i$ and centers positioned at $\mathbf{r} = \mathbf{r}_i$. In addition to these two fields one also needs to introduce an incident field, which represents the initial incident field (first term in the equation below) and the fields scattered by all other disks (the second term):

$$B_{inc}^{(i)} = \sum_m a_m^{(i)} e^{im\varphi_i} J_m \left( n_0 k |\mathbf{r} - \mathbf{r}_i| \right) + \sum_{j \neq i} \sum_m b_m^{(j)}(\omega) e^{im\varphi_j} H_m \left( n_0 k |\mathbf{r} - \mathbf{r}_j| \right)$$

(17)

Here polar angles $\varphi_j$ and radial coordinates are defined in local coordinate systems associated with each $j-th$ disk. For the scattering problem considered here arguments of all Bessel functions contain the frequency of the incident field. In order to be able to use boundary conditions at the rim of the $i-th$ disk the scattered field of all disks needs to be rewritten in the coordinate system centered at the $i-th$ disk. This is achieved with the help of the Graffs formula for the Hankel function [5]

$$e^{im\varphi_j} H_m \left( n_0 k |\mathbf{r} - \mathbf{r}_j| \right) = \sum_{n=-\infty}^{n=\infty} H_{n-m} \left( n_0 k R_{ji} \right) e^{i(m-n)\theta_{ji}} J_n \left( n_0 k |\mathbf{r} - \mathbf{r}_i| \right) e^{im\varphi_i}$$

(18)

where $R_{ji}$ and $\theta_{ji}$ are radial and polar coordinates of the $i-th$ disk in the coordinate system centered at $j-th$ disk. Using this theorem a single disk results is generalized into the multiple disk form

$$b_m^{(i)}(\omega) = \alpha_m^{(i)}(\omega) \left[ a_m^{(i)} + \sum_{j \neq i} \sum_n b_n^{(j)}(\omega) H_{m-n} \left( n_0 k R_{ji} \right) e^{i(n-m)\theta_{ji}} \right]$$

(19)

where $\alpha_m^{(i)}(\omega)$ is a single disk scattering coefficients

$$\alpha_m^{(i)} = \frac{n_d J_m(kn_d R_i) J_m'(kR_i) - J_m(kR_i) J_m'(kn_d R_i)}{H_m(kR_i) J_m'(kn_d R_i) - n_d J_m(kn_d R_i) H_m'(kR_i)} \tag{20}$$

In the case of a double-disk system Equation (19) simplifies into

$$b_m^{(1)}(\omega) = \alpha_m^{(1)} \left[ a_m^{(1)} \delta_{mM} + \sum_n (-1)^{n-m} b_n^{(2)} H_{m-n}(kD) \right]$$

$$b_m^{(2)}(\omega) = \alpha_m^{(2)} \sum_n b_n^{(1)}(\omega) H_{m-n}(kD) \tag{21}$$

where $D$ is the distance between the disks' centers, and it was assumed that only the $M$-th mode of the first disk is being excited by an original incident field. This is the system of infinite number of equations, which describes coupling between different azimuthal modes. These equations were analyzed under assumption that all disks are identical, which is a very good approximation for the structures under consideration. To solve these equations it should be first noticed that the strongest coupling takes place between modes with equal or close single disk resonances. Since in microdisks modes with azimuthal numbers of the same magnitude but opposite signs are degenerate, one first needs to consider coupling between modes described by coefficients $b_{\pm M}^{(i)}$. Also taking into account that the Hankel functions grow fast with increasing order, one realizes that coupling between coefficients $b_M^{(1)}$ and $b_{-M}^{(2)}$, characterized by coupling parameter $H_{2M}(kD)$, is much stronger than the coupling between coefficients $b_M^{(1)}$ and $b_M^{(2)}$, which is characterized by $H_0(kD)$. Thus, one can solve Equation (21) iteratively, taking first into account the coupling between main modes, and incorporating the rest using perturbation theory with coupling coefficients $H_{M-n} / H_{2M}$ as small parameters. The zero order approximation takes into account only coupling between counter-propagating resonant modes with following analytical result for new resonances modified by coupling:

$$\left[ \alpha_M(k) \right]^{-1} = \pm H_{2M}(kD) \tag{22}$$

The main feature of this approximation is that it preserves degeneracy between two modes, one of which can be described as symmetric and characterized by $b_M^{(1)} + b_{-M}^{(2)}$ combination of the coefficients, while the other one is antisymmetric characterized by $b_M^{(1)} - b_{-M}^{(2)}$ combination. This equation gives a simple method to estimate the effect of coupling on normal modes and was used

in conjunction with experimental results to verify the effieiency of coupling in the fabricated structures.

This approximation was further improved by taking into account coupling to the non-resonant modes in the first order of the perturbation theory. The results can be presented in the following form

$$\left[\alpha_M(k)\right]^{-1} = \pm H_{2M}(kD) + \sum_{n \neq -M} \alpha_n \left[H_{M-n}(kD)\right]^2 \pm \sum_{n \neq -M} \alpha_n^{(i)} H_{M+n}(kD) H_{n-M}(kD) \quad (23)$$

which shows that now there are four distinct resonance frequencies so that the degeneracy between symmetric and antisymmetric modes is now removed. Using this result one can determine if the structure under consideration requires going beyond the resonant approximation for its modeling. Since in our experiments only two frequencies have been observed we can conclude that interaction with non-resonant modes is not important and can be neglected.

## 2.4. Summary

Theoretical and modeling efforts with this Phase I project produced following results, which are crucial for the successful completion of the next phase of this project:

1.  We ascertained the physical mechanism responsible for multistable lasing in the system of two waveguides coupled through a resonant element (ring or disk resonator)

2.  The developed model of lasing in this type of structures allowed predicting emission properties of the structure for given morphological characteristics and tailor the structural parameters to achieve desirable multistable properties

3.  For the system of coupled disks a theoretical model and necessary computer codes were developed for simulating properties of normal modes of these structures. The obtained results are crucial for further modeling of lasing from these structures.

## 3.0     EXPERIMENTAL

The experimental work performed under this Phase I grant can be divided into two main subtasks:

1.  The microring integrated with an active waveguide system and

2.  The coupled active microdisk system

Both these systems were investigated for their bi-stable output characteristics in addition to their linear transmission and luminescence properties.

## 3.1. Microring integrated with an active waveguide

As part of this task, we have been involved in the design of the all optical flip-flop. The schematic drawing of the proposed architecture is shown in Figure 6. Here two gain media are integrated with a single microring resonator as shown in the schematic drawing. Specific subtasks investigated are briefly described below.
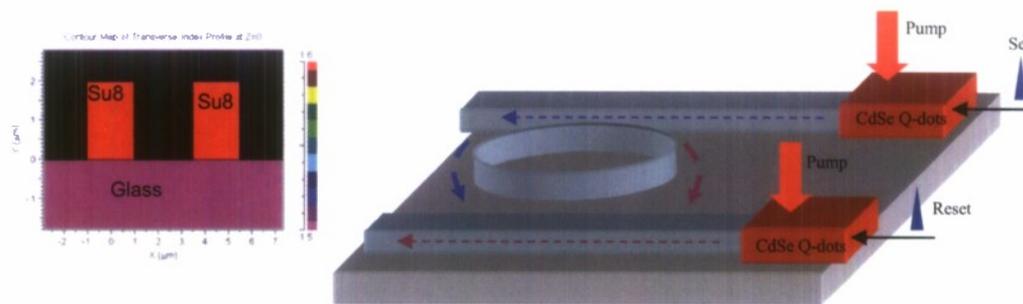


**Figure 6: Schematic Drawing of the Proposed All-optical flip flop which uses a passive microring resonator integrated with active elements. Also shown is the crossesction of the SU8 waveguides and the glass substrate. The waveguides are 2 μm×2μm in size**

### 3.1.1. Design of ring resonator

Firstly we investigated the bend loss mechanism in a waveguide made using SU8 – a negative photosensitive polymer. Simulations were performed using BEAMPROP, a commercially available waveguide modeling tool. Shown in Figure 7 is calculated loss of a SU8 waveguide (2 μm × 2 μm) at different wavelengths for a 180 degree bend.
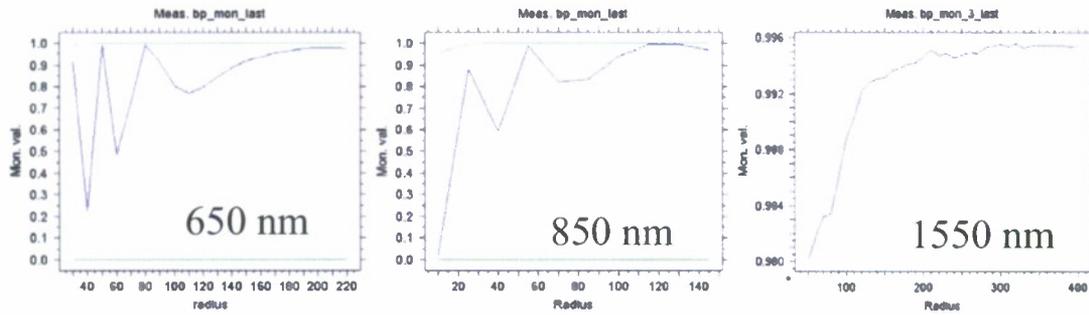
**Figure 7  Calculated Bending Loss for Different Input Wavelengths as a Function Of Bend Radius for a 180 Degree Bend**

The blue line in the plots above indicates the power in the fundamental mode as a function of bending radius for different input wavelength values. The optimal values of radii chosen for the different wavelengths were 205, 130, and 100 μm.

### 3.1.2.   Design of Multimode Interference Coupler.

We simulated the performance of the coupler which will be used for coupling the light from the straight waveguides to the ring resonator using beam propagation software. Results of simulations are shown in Figure 8.
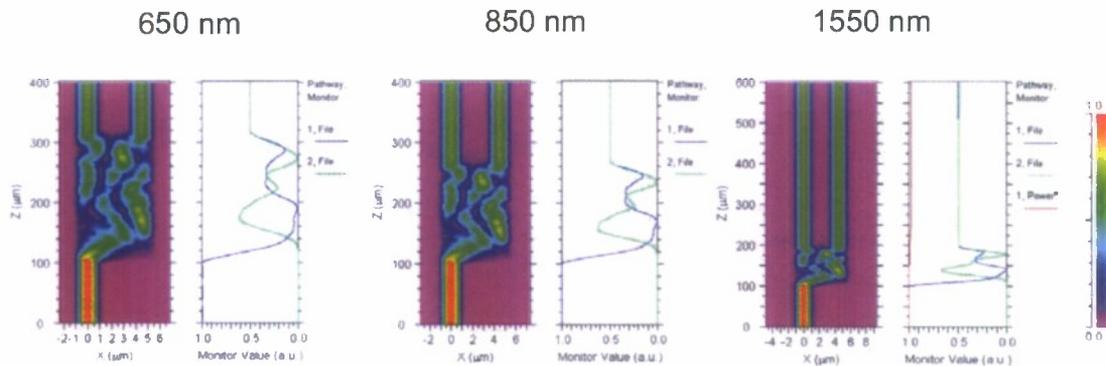


**Figure 8: Simulations of Multi-mode Interference Couplers with 50-50 Splitting Ratio for Different Input Wavelengths**

The couplers were based on multi-mode interference (MMI) scheme and were designed for 50-50 power splitting ratio.

14

### 3.1.3. Resonant frequencies of the ring resonator with MMI couplers

Following the design of the MMI coupler and the calculations of optimal bend radius, we simulated the resonant frequencies of the ring resonator. To this end we calculated transmission coefficient of a coupler, $T$, using a model, in which the coupling to the resonator is described by a $2 \times 2$ transfer matrix, connecting fields incident at the coupler from the waveguide, and from the ring. The coupler is characterized by power coupling parameters $\kappa$, and the loss parameters $\gamma$, and $\alpha$, where former characterizes the fraction of power lost at the coupler, while the latter takes into account absorption in the resonator. This approach, which is standard for MMI couplers and is used in a number of works, results in the following expression for the power transmission coefficients defined as a fraction of energy propagating in the waveguide passing the coupler:

$$T = (1-\gamma) \left| \frac{(1-\kappa)^{1/2} - (1-\gamma)^{1/2} \exp\left(-\frac{1}{2}\alpha L - i\Phi\right)}{1 - (1-\kappa)^{1/2}(1-\gamma)^{1/2} \exp\left(-\frac{1}{2}\alpha L - i\Phi\right)} \right|^2 \qquad (24)$$

where L is the circumference of the resonator, and $\Phi = \beta_r L$, where $\beta_r$ is the propagation constant in the ring. Figure 9 shows the transmission characteristics of the ring resonator at different wavelengths. The minima in transmission correspond to the resonance coupling of light to the resonator.
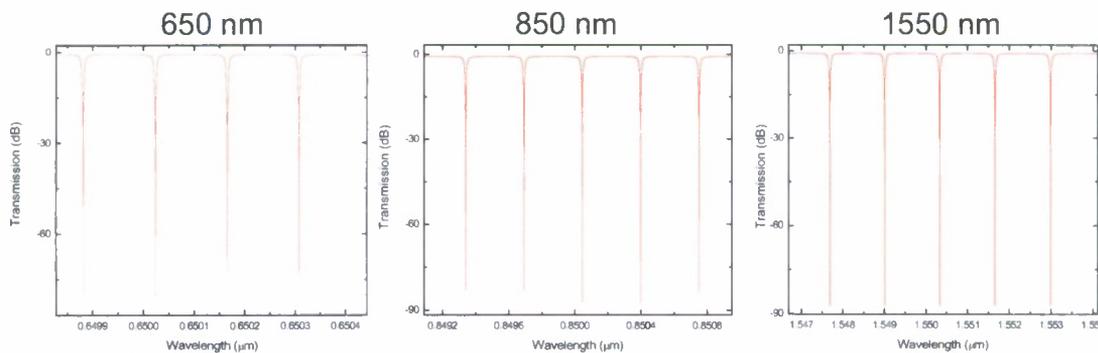


**Figure 9: Simulations of Resonant Frequencies of The Ring Resonators Designed for Three Different Wavelength Ranges : 650, 850, and 1550 nm**

### 3.1.4.  Fabrication of microring structure

Microring structures were realized using both standard photolithography as well as electron beam lithography following the design parameters established above. One of the key components in this architecture is the integration of active waveguide consisting of colloidal quantum dots (CdSe/ZnS) embedded in a polymer host with a passive waveguide. We have successfully implemented this step using a multilayer electron beam lithography process with appropriate alignment marks. Scanning electron microscope image of an active waveguide integrated with a passive waveguide is shown in Figure 10 (a). Also shown in the same figure is the scanning electron microscope image of the microring resonator (b).
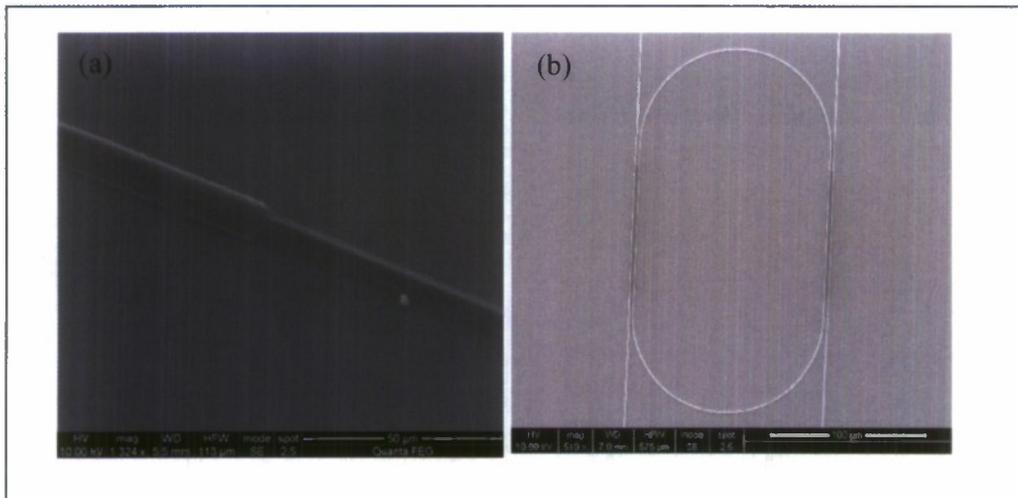


**Figure 10:  Scanning Electron Microscope Images Of (A) Active Waveguide Integrated With Passive Waveguide, And (B) Microring Resonator Coupled To Passive Waveguides. Here, The Active Waveguides Consist Of Colloidal Cdse Qds Embedded In A SU8 Matrix.**

The active passive integrated waveguide structure was characterized by optical pumping of the device using an Argon ion laser (488 nm). During the initial phase of device fabrication, one of the major issues we had was that the colloidal quantum dots were getting deposited all over the wafer despite removing the polymer using the developer. This issue was overcome and now we have been able to realize devices where only the active region has the quantum dots. Shown in

Figure 11 is an optical microscope image of an active waveguide integrated with a passive waveguide along with the image under optical excitation where the red emission is observed only from the active region.
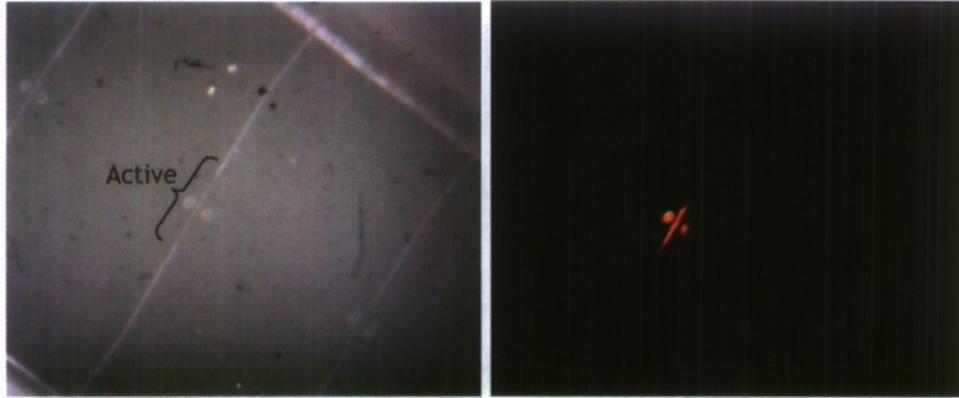


**Figure 11: Optical Microscope Image of the Active Waveguide Integrated with a Passive Waveguide (a) and Image of Emission from the Active Waveguide under Optical Excitation (b)**

The ring resonator was characterized for its transmission properties using a broadband source operating in the 1550 nm wavelength range. Although the active material we had chosen was CdSe quantum dots that emit in the red (620 nm), the ease of characterizing in the infrared prompted us to do the transmission measurements in the infrared. Shown in Figure 12 is one of the transmission resonances of the ring resonator in the 1550 nm wavelength range.
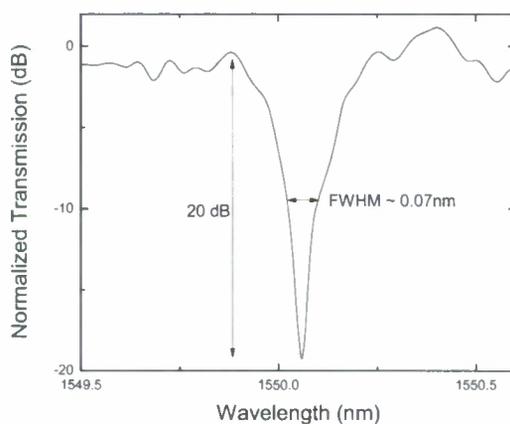
**Figure 12: Transmission Characteristics of the Ring Resonator Shown Indicating Extinction of ~20 dB and 0.07 nm FWHM**

**Summary:** We have successfully fabricated a microring resonator integrated with two active waveguides. The integration of the active waveguide with the passive waveguide was shown to be successful based on the emission characteristics. Although we have not yet observed bistable lasing behavior from this integrated device, experiments are currently underway to demonstrate this in the near future. The outstanding issues include low reflectivity from the waveguide facets, and losses in the ring resonator.

### 3.2. Coupled Active Microdisks

In this approach towards realizing bistable lasing, we use active microdisks consisting of CdSe/ZnS colloidal uantum dots embedded in a photosensitive polymer. In the present demonstration, the polymer used was SU8 (Microchem Corp.), a negative photoresist. Patterning of the microdisks were achieved using both soft-lithography and electron beam lithography. Due to the resolution limit of soft-lithography, it was not possible to realize efficient coupled microdisks using this technique. Hence we adopted electron beam lithography to realize this structure. Scanning electron microscope image of the coupled microdisks embedded with CdSe quantum dots is shown in Figure 13.
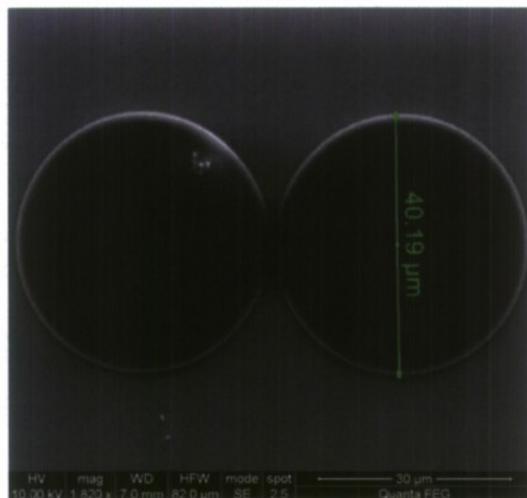
**Figure 13: Scanning electron microscope image of coupled microdisks fabricated using electron beam lithography. The disks are 40 μm in diameter and are separated by 300 nm**

Here the separation between the disks was varied from 250 to 350 nm to tune the coupling between them. These coupled microdisks were characterized for their emission properties. Shown in Figure 14 is the image of emission observed using a CCD camera from a three coupled microdisk structure along with the optical microscope image of the sample under investigation.
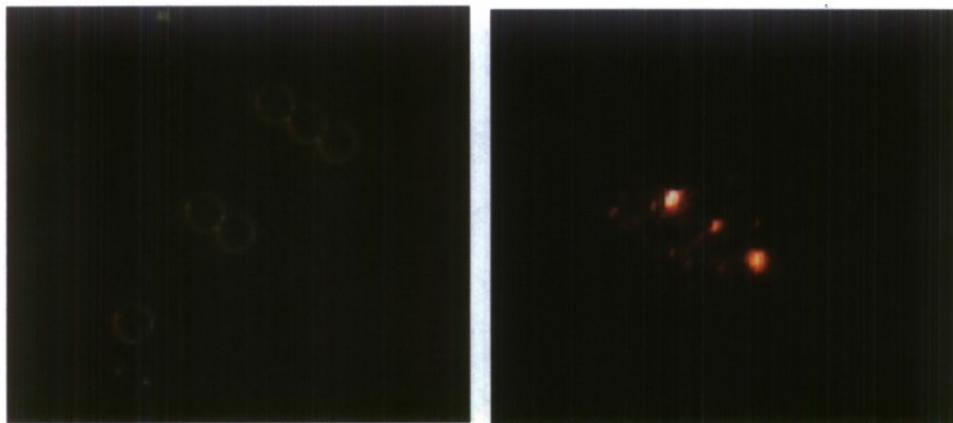


**Figure 14: Optical Microscope Image of The Coupled Microdisk Structure Along with the Emission from the Three Coupled Microdisk Sample**

We also performed spectral measurements on the coupled microdisks. The experimental set up used to characterize these microdisks is shown in Figure 15.
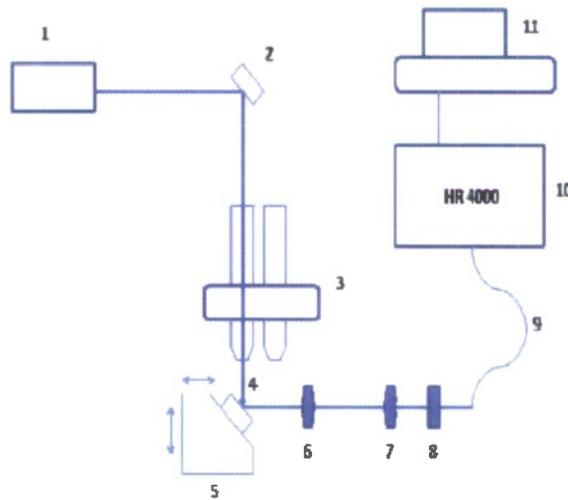


**Figure 15: The experimental setup. 1 – laser, 2 – mirror, 3 - microscope, 4 – sample, 5 – 3D stage with rotation, 6 – f= 25 mm lens, 7 - f= 15 mm lens, 8 – filter, 9 – optical fiber, 10 – HR 4000 CG-UV-NIR spectrometer, 11 –computer.**

The spectral property of the coupled microdisks under optical excitation using an Argon-ion laser was studied using a high resolution spectrometer. Optical spectrum observed from a two coupled microdisk sample is shown in Figure 16. Two clear peaks are observed above the spontaneous emission spectrum. While we have not yet confirmed if these peaks are indeed lasing peaks, the significantly narrower linewidths indicate that there is gain occurring in these systems. In addition, bright emission observed from these disks further substantiates our claims.
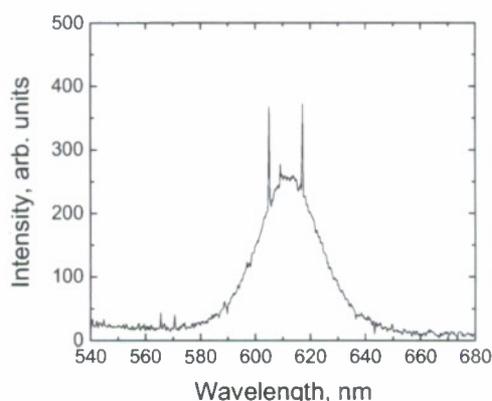
**Figure 16: Emission from a Two Coupled Microdisk Structure Showing Two Significantly Narrow Peaks**

**Summary:** We have successfully demonstrated coupled microdisks using colloidal quantum dot composites. These disks have demonstrated excellent spectral properties and have shown spectrally selective emission. Outstanding issues to be addressed include demonstration of optical control of bistable emission, integration of passive waveguides to these active disks, and short operational lifetime of these active microdisks.

## 4.0    CONCLUSIONS AND RECOMMENDATIONS

Under the Phase I grant, the Hybrid Photonics + Queens College team have successfully performed feasibility study on realizing bistable optical devices using novel physical mechanisms. Specific tasks accomplished include:

- Developed the theory of bistable lasing in a system consisting of a single resonator coupled to waveguides.
- Developed software that predicts the lasing modes in such systems
- Successfully fabricated microring resonator side coupled to active waveguides
- Successfully integrated colloidal quantum dot based active waveguide with a passive waveguide.
- Developed theoretical model and necessary computer codes for simulating properties of normal modes of coupled micridisk structures. The obtained results are crucial for further modeling of bistable lasing from these structures.

- Fabricated coupled microdisk structures and demonstrated spectrally selective narrowband multiwavelength emission.

Recommendations for future work:

Based on our preliminary results of Phase I, the coupled active microdisk resonator system looks more promising than the ring resonator system to obtain bistable lasing. Theoretical simulations have clearly indicated the possibility to obtain bistable lasing in the former system. Furthermore, experimentally we have observed narrow spectral emission from these coupled microdisk structures. Both theory and experiment favor the former system for realizing

In future work, we will use the coupled microdisk resonator system to realize the all-optical flip FF. The choice of this architecture for realizing the FF is based on recent theoretical and experimental results we have obtained. The ring-resonator based structure was considered as a quick first step in realizing a principally new type of bistability providing spectral diversity, but the progress in this direction was hindered by unanticipated difficulties such as low light intensities in the passive waveguides. Nevertheless, as the main idea of this device does not depend on specific [ring-resonator] architecture, we have found recently that the device is realized easier by microdisk structures. Thus we will continue pursuing an all-optical FF based on the idea of the resonance coupling of active waveguides by a microdisk resonator. It is important to point out that the study of the waveguide coupled single-disk structure is, by any means, a necessary intermediate step toward realizing coupled-disk structures. Thus, while the main focus of our future work will be on coupled-disk based optical flip-flops, we anticipate realizing the principally new type of single-disk FF en route toward the main goal. The proposed modifications in our future efforts reflect realization that the development of a double-disk based FFs has lesser risks as well as guarantees excellent chances of its successful realization.

## REFERENCES

1. Sargent III, M., Scully, M. O., Lamb, Jr., W. E. **Laser Physics**, Westeview Press, 1974, p. 96 – 113

2. Borselli, M., Johnson, T. J., Painter, O., "Beyond the Rayleigh scattering limit in high-Q silicon microdisks: theory and experiment", *Optics Express*, **13**, 5, February 2005, pp. 1515 - 1530

3. Türeei, H. E., Ge, L., Rotter, S. Stone, A. D., "Ab initio self-eonsistent laser theory and random lasers," *Nonlinearity*, **22**, Deeember 2008 pp. C1 – C18.

4. Ching, E. S. C., Leung, P. T. Maassen van den Brink, A., Suen, W. M., Tong, S. S., Young, K., *Quasinormal-mode expansion for waves in open systems*, Rev. Mod. Phys., **70**, 4, Oetober 1998, pp. 1545 – 1554.

5. Boriskina S. V., *Spectrally engineering photonic molecules as optical sensors with enhanced sensitivity: a proposal and numerical analysisI,* J. Opt. Soe. Am. B, **23**, 8, August 2006, pp. 1565 – 1573.

# APPENDIX A

## CALCULATIONS OF NORMAL AND SCATTERING RESONANCES FOR INDIVIDUAL AND COUPLED DISKS

### Scattering resonances:

```
n0 = 0; nd = 0; (* to make global, do not change it *)
DiskTE[mmin_, mmax_, mpoints_, kmin_, kmax_, kpoints_, levels2count_] :=
   Module[{out = Table[{}, {6}], x1, a1, j, ReK, ImK, gImK, m, Nstep, startTime,
     endTime, g, levels0 = levels2count, countLevels, R = SetPrecision[R, myPrec]},
   Off[General::unfl];
   If[levels2count == -1, levels0 = 10^5;];
   x1 = SetPrecision[Range[kmin, kmax, (kmax - kmin) / kpoints], myPrec];
   startTime = AbsoluteTime[];
   If[stepsN ≠ 0, Nstep = deltaN / stepsN, Nstep = {0, 0};];
   Do[ (*3*)
         countLevels = 0;
          Print["Working with order=", m, "   ", DateString[]];
         n0 = SetPrecision[nInd[[1]], myPrec];
         nd = SetPrecision[nInd[[2]], myPrec];
         a1 = fZeroImag[x1, m, R];
         Do[ (*2*)
          If[countLevels >= levels0,
            Break[];
            ];
            If[a1[[g]] * a1[[g + 1]] < 0,
            Do[ (*1*)
             If[j == 0,
               n0 = SetPrecision[nInd[[1]], myPrec];
               nd = SetPrecision[nInd[[2]], myPrec];
               ZeroImag = FindRoot[fZeroImag[x, m, R], {x, x1[[g]]}, AccuracyGoal →
               myGoals, PrecisionGoal → myGoals / 2, WorkingPrecision → myPrec - 1];
               ReK = ZeroImag[[1]][[2]];
               ImK = ReK / 2 * n0 * nd * (-BesselJ[m, ReK * nd * R] *
                   (BesselJ[m - 1, ReK * n0 * R] - BesselJ[m + 1, ReK * n0 * R]) + BesselJ[m,
                    ReK * n0 * R] * (BesselJ[m - 1, ReK * nd * R] - BesselJ[m + 1, ReK * nd * R]));
```

```
        gImK = 1 / (ReK * R) * (2 * ReK * (n0 - nd) * nd * R * BesselJ[m - 1, ReK * nd * R] *
          (ReK * n0 * R * BesselY[m - 1, ReK * n0 * R] -
            m * BesselY[m, ReK * n0 * R]) + BesselJ[m, ReK * nd * R] *
          (ReK * n0 * R * (-ReK * n0 * nd * R * BesselY[m - 2, ReK * n0 * R] - 2 * (m * n0 + nd -
                  2 * m * nd) * BesselY[m - 1, ReK * n0 * R]) + (4 * m^2 * (n0 - nd) +
              ReK^2 * n0 * (n0 - 2 * nd) * nd * R^2) * BesselY[m, ReK * n0 * R]));
      startRe = ReK;
      startIm = ImK / gImK;

      '
      If[Nstep[[1]] == 0 || deltaN[[1]] == 0,
        n0 = SetPrecision[nInd[[1]], myPrec];

        '
        n0 = SetPrecision[nInd[[1]] + I * j * Nstep[[1]], myPrec];];
  (*complex part of n0 *)
      If[Nstep[[2]] == 0 || deltaN[[2]] == 0,
        nd = SetPrecision[nInd[[2]], myPrec];

        '
        nd = SetPrecision[nInd[[2]] + I * j * Nstep[[2]], myPrec];];
  (*complex part of nd *)
       startRe = X[[1]];
     startIm = X[[2]];
      ];
    value = FindMinimum[scatter[rek, imk, m, R], {rek, SetPrecision[startRe,
     myPrec]}, {imk, SetPrecision[startIm, myPrec]}, AccuracyGoal → myGoals,
  WorkingPrecision → myPrec, Method → "PrincipalAxis"];
    X = {value[[2]][[1]][[2]], value[[2]][[2]][[2]]};
    Y = value[[1]]; (* absolute function value *)
    , {j, 0, stepsN}]; (*/l*)
 countLevels ++;
out[[1]] = Append[out[[1]], X[[1]] * R];
out[[2]] = Append[out[[2]], X[[2]] * R];
out[[3]] = Append[out[[3]], Y];
out[[4]] = Append[out[[4]], m];
out[[5]] = Append[out[[5]], X[[1]] / Abs[X[[2]]]];
out[[6]] = Append[out[[6]], countLevels];  (*number of the level*)
 ];
```

```
      , {g, 1, kpoints - 1}]; (*/2*)
        , {m, mmin, mmax, mpoints}];   (*/3*)
      endTime = AbsoluteTime[];
      Print["Execution time:", (endTime - startTime), " seconds"];
      If[out[[1]] == {},
        Print["No solutions found, increase max K value or number of K points"];];
      out (* return array *)
    ];
  fZeroImag[x0_, m0_, R_] := Module[{k0, kd, m = SetPrecision[m0, myPrec], x = x0, val},
      k0 = x * n0;
      kd = x * nd;
      val := k0 * nd * BesselJ[m - 1, kd * R] * BesselY[m, k0 * R] + 1 / R * BesselJ[m, kd * R] *
          (-k0 * R * nd * BesselY[m - 1, k0 * R] + m * (nd - n0) * BesselY[m, k0 * R]);
      val
    ];
  scatter[RK_, IK_, m0_, R_] := Module[{k0 = SetPrecision[(RK + I * IK) * n0, myPrec],
        kd = SetPrecision[(RK + I * IK) * nd, myPrec], val, m = SetPrecision[m0, myPrec]},
      val := N[n0 * HankelH1[m, k0 * R] * kd / 2 * (BesselJ[m - 1, kd * R] - BesselJ[m + 1, kd * R]) -
          nd * BesselJ[m, kd * R] * k0 / 2 *
            (HankelH1[m - 1, k0 * R] - HankelH1[m + 1, k0 * R]), myPrec];
      Sqrt[Re[val]^2 + Im[val]^2]
    ];
  (* supplementary functions *)
  myFileName := "scat." <> ToString[nInd[[1]]] <> "(" <> ToString[deltaN[[1]]] <> ")_" <>
      ToString[nInd[[2]]] <> "(" <> ToString[deltaN[[2]]] <> ")_" <> ToString[R];
  plotResults[res_, box_, saveFormat_, useColor_] := Module[{plStyle, gr, fname,
        t = Table[{}, {4}], styles = Table[{}, {4}], modes, temp, k, color},
      If[res[[1]] == {}, Return[];];
      (*Off[Show::shx];*) (* hide warning for empty graph *)
      plStyle = {PlotRange -> All, PlotMarkers -> "●", Frame -> True,
          ImageSize -> {400, 300}, LabelStyle -> {FontSize -> 14}, Axes -> False};
      styles[[1]] = {plStyle, FrameLabel -> {"Re(kR)", "Mode"}};
      styles[[2]] = {plStyle, FrameLabel -> {"Re(kR)", "Im(kR)"}};
      styles[[3]] = {plStyle, FrameLabel -> {"Re(kR)", "Q-factor"}};
      styles[[4]] = {plStyle, FrameLabel -> {"Re(kR)", "Absolute Value"}};
      If[!useColor,
```

```
                t[[1]] = ListPlot[Transpose[{res[[1]], res[[4]]}], styles[[1]]];
                t[[2]] = ListPlot[Transpose[{res[[1]], res[[2]]}], styles[[2]]];
                t[[3]] = ListLogPlot[Transpose[{res[[1]], res[[5]]}], styles[[3]]];
                t[[4]] = ListLogPlot[Transpose[{res[[1]], res[[3]]}], styles[[4]]];
        , (*else*)
                modes = Max[res[[6]]];
                Do[
                        temp = Transpose[Select[Transpose[res], #[[6]] == k &]];
                        color = ColorData["VisibleSpectrum"][380 + (750 - 380) * (k - 1) / modes];
                        t[[1]] = Show[t[[1]], ListPlot[Transpose[{temp[[1]], temp[[4]]}],
                {styles[[1]], PlotStyle → color}], PlotRange → All];
                        t[[2]] = Show[t[[2]], ListPlot[Transpose[{temp[[1]], temp[[2]]}],
                {styles[[2]], PlotStyle → color}], PlotRange → All];
                        t[[3]] = Show[t[[3]], ListLogPlot[Transpose[{temp[[1]], temp[[5]]}],
                {styles[[3]], PlotStyle → color}],
                PlotRange → All, FrameTicks → {Automatic, Automatic}];
                        t[[4]] = Show[t[[4]], ListLogPlot[Transpose[{temp[[1]], temp[[3]]}],
                {styles[[4]], PlotStyle → color}],
                PlotRange → All, FrameTicks → {Automatic, Automatic}];
                        , {k, modes}];
        ];
        If[box || StringQ[saveFormat],
          gr = GraphicsGrid[{{t[[1]], t[[2]]}, {t[[3]], t[[4]]}}];,
          gr = GraphicsColumn[{t[[1]], t[[2]], t[[3]], t[[4]]}];
            ];
        If[StringQ[saveFormat],
          fname = myFileName <> "." <> saveFormat;
          Print["Saving graphics to " <> fname];
          Export[NotebookDirectory[] <> fname, gr]
          (* "Image" is not required and produces strange image file,
          but there is a bug in Mathematica 7 - it can not export ListLogPlot *)
          ];
        Print[gr];
    ];
(* saving data to the file *)
saveData[data_, lim_] := Module[{k, l, fname, res, uniqueVals, saveData, pos, f},
```

```
If[data[[l]] == {}, Return[];];
(* check same value *)
uniqueVals = Union[data[[l]], SameTest -> ((N[#1, lim] == N[#2, lim]) &)];
If[Length[data[[l]]] != Length[uniqueVals],
  Print["There are the same solutions. Try to perform more accurate calculations
      (starting points).\nThese values for Real part are:"];
  Print[Commonest[N[data[[l]], 12]]];
  Print["They will be removed from saved data"];
  saveData = Table[0, {Length[data]}, {Length[uniqueVals]}];
  For[k = 1, k <= Length[uniqueVals], k++,
   pos = Position[N[data[[l]], lim], N[uniqueVals[[k]], lim]];
   For[l = 1, l <= Length[data], l++,
     saveData[[l]][[k]] = data[[l]][[pos[[l]][[1]]]]];
  ];
  , (*if ... else*)
  saveData = data;
];
res = Transpose[saveData];
fname = myFileName <> ".txt";
f = OpenWrite[NotebookDirectory[] <> fname ];
Do[
  WriteString[f, StringJoin[
       Riffle[ToString[FortranForm[N[#, lim]]] & /@ res[[i]], "\t"]] <> "\n"];
  , {i, 1, Length[res]}];
Close[f];
Print["Data saved to ", fname,
  " in the format Re(kR), Im(kR), Abs Value, Mode, Qfactor"];
];
```

```
(* main part with the structure parameters *)
myPrec := 50; (* working precision. should
 be larger than myGoals in approximately two times *)
myGoals := 16; (* required precision for solution *)
nInd := {1, 1.5}; (* n0 and nd *)
R := 60;  (* radius *)
stepsN := 10;  (* number of steps for changing index of refraction *)
deltaN := {0, 0.00001}; (* change in complex part of the index of refraction *)
(* calling parameters: m_min, m_max, k_min, k_max,
k_points, number of levels to find (-1 for 10^5 levels) *)
results = DiskTE[5, 15, 10, .1, 2.1, 250, 3];
saveData[results, 16]; (* data, number of digits to save into file *)
plotResults[results, True, "gif", False] (* data,
2x2 (True) or 1x4 (False) output, Export graphics to "format"("gif","jpg",
   "eps".. check manual for Export supported formats) file (False to disable),
use different color for each level (True) *)
```

## Normal resonances:

```
n0 = 0; nd = 0; (* to make global, do not change it *)
DiskModes[kmin_, kmax_, kpoints_, kDmin_, kDmax_, kDpoints_, levels2find_] :=
  Module[{R = SetPrecision[R, myPrec], ReK, ImK, gImK, startTime, endTime, j, g1, g2, k, x0,
    x1, levels0 = levels2find, out = Table[{}, {6}], m = SetPrecision[mode, myPrec],
    countLevels, Nstep, debug = False, myBreak, pts, ZeroImag, a1, rek, imk, same},
  Off[General::unfl];
  If[levels2find == -1, levels0 = 10^5;];
  x0 = SetPrecision[Range[kmin, kmax, (kmax - kmin) / kpoints], myPrec];
  x1 = SetPrecision[Range[kDmin, kDmax, (kDmax - kDmin) / kDpoints], myPrec];
  startTime = AbsoluteTime[];
  If[stepsN ≠ 0, Nstep = deltaN / stepsN, Nstep = {0, 0};];
  Do[ (*3*)
        countLevels = 0;
        k = x0[[g1]]; (* value of the continuous K *)

   Print["Working with k: ", N[k, 6], " ", g1, " out of ", kpoints, " ", DateString[]];
        n0 = SetPrecision[nInd[[1]], myPrec];
        nd = SetPrecision[nInd[[2]], myPrec];
        a1 = fZeroImag[x1, m, k];
        Do[ (*2*)
         If[countLevels >= levels0,
          Break[];
          ];
          If[a1[[g2]] * a1[[g2 + 1]] < 0,
           myBreak = False;
          Do[ (*1*)
           If[j == 0,
             n0 = SetPrecision[nInd[[1]], myPrec];
             nd = SetPrecision[nInd[[2]], myPrec];
             ZeroImag = FindRoot[fZeroImag[x, m, k], {x, x1[[g2]], x1[[g2 + 1]]}, Method →
            "Brent", AccuracyGoal → myGoals / 2, WorkingPrecision → Round[myPrec / 1.2]];
             ReK = ZeroImag[[1]][[2]];
             ImK = 1 / 2 * n0 * nd * (k^2 * BesselJ[m, ReK * nd * R] *
               (BesselJ[m - 1, k * n0 * R] - BesselJ[m + 1, k * n0 * R]) + ReK^2 * BesselJ[m,
               k * n0 * R] * (BesselJ[m + 1, ReK * nd * R] - BesselJ[m - 1, ReK * nd * R]));
             gImK = 1 / 2 / R * (k^2 * n0 * nd^2 * R^2 * (BesselJ[m - 1, ReK * nd * R] -
                   BesselJ[m + 1, ReK * nd * R]) * BesselY[m - 1, k * n0 * R] +
             (-2 * nd * (ReK * n0 + k * m * nd) * R * BesselJ[m - 1, ReK * nd * R] + 2 / ReK *
                   (-ReK * (m - 1) * m * n0 + k * m^2 * nd + ReK^3 * n0 * nd^2 * R^2) *
                   BesselJ[m, ReK * nd * R]) * BesselY[m, k * n0 * R]);
```

```
        startRe = ReK;
        startIm = ImK / gImK;

        ,
        If[Nstep[[1]] == 0 || deltaN[[1]] == 0,
          n0 = SetPrecision[nInd[[1]], myPrec];

            ,
          n0 = SetPrecision[nInd[[1]] + I * j * Nstep[[1]], myPrec];];
    (*complex part of n0 *)
        If[Nstep[[2]] == 0 || deltaN[[2]] == 0,
          nd = SetPrecision[nInd[[2]], myPrec];

            ,
          nd = SetPrecision[nInd[[2]] + I * j * Nstep[[2]], myPrec];];
    (*complex part of nd *)
          startRe = X[[1]];
        startIm = X[[2]];
        ];
      If[debug,
          Print["Starting from:", startRe, " and ", startIm];
          Print["range ", x1[[g2]], " - ", x1[[g2 + 1]]];
       pts = Reap[FindMinimum[scatter[rek, imk, k, m],
        {rek, SetPrecision[startRe, myPrec]}, {imk, SetPrecision[startIm, myPrec]},
        AccuracyGoal → myGoals, WorkingPrecision → myPrec,
        Method → "PrincipalAxis", StepMonitor :→ Sow[{rek, imk}]]][[2, 1]];
          Print[value];
          Print[pts];
        ];
      value = FindMinimum[scatter[rek, imk, k, m],
     {rek, SetPrecision[startRe, myPrec]}, {imk, SetPrecision[startIm, myPrec]},
    AccuracyGoal → myGoals, WorkingPrecision → myPrec, Method → "PrincipalAxis"];

      X = {value[[2]][[1]][[2]], value[[2]][[2]][[2]]};
      Y = value[[1]]; (* absolute function value *)
      If[X[[1]] < kDmin, Print["Point skipped - solution found below the min k_m"];
  myBreak = True; Break[];]; (* break of solution found below the range *)
       , {j, 0, stepsN}]; (*/l*)
(* check whether these values were found before *)
same = Length[Select[N[Transpose[out], 10], (#[[1]] == N[X[[1]] * R, 10] &&
      #[[2]] == N[X[[2]] * R, 10] && #[[4]] == N[k * R, 10]) &]];
 If[same == 0, countLevels++, Print["Solution found before"]];
    If[! myBreak && same == 0,
       out[[1]] = Append[out[[1]], X[[1]] * R];
       out[[2]] = Append[out[[2]], X[[2]] * R];
```

```mathematica
                    out[[3]] = Append[out[[3]], Y];
                    out[[4]] = Append[out[[4]], k * R];
                    out[[5]] = Append[out[[5]], X[[1]] / Abs[X[[2]]]];
                    out[[6]] = Append[out[[6]], countLevels]; (*number of the level*)
                    ];
                ];
          , {g2, 1, kDpoints}]; (*/2*)
        , {g1, 1, kpoints}]; (*/3*)
      endTime = AbsoluteTime[];
      Print["Execution time:", (endTime - startTime), " seconds"];
      If[out[[1]] == {},
        Print["No solutions found, increase max K value or number of K points"];];
      out (* return array *)
    ];
fZeroImag[x0_, m_, k_] := Module[{val, x = x0, R = SetPrecision[R, myPrec]},
    val := SetPrecision[-x^2 * n0 * nd * BesselJ[m - 1, x * nd * R] * BesselY[m, k * n0 * R] +
        BesselJ[m, x * nd * R] * (k^2 * n0 * nd * BesselY[m - 1, k * n0 * R] +
            m / R * (x * n0 - k * nd) * BesselY[m, k * n0 * R]), myPrec];
    val
    ];
scatter[RK_, IK_, k_, m_] :=
  Module[{x = SetPrecision[(RK + I * IK), myPrec], val, R = SetPrecision[R, myPrec]},
    val := SetPrecision[n0 * x * HankelH1[m, k * n0 * R] * 1 / 2 * x * nd *
        (BesselJ[m - 1, x * nd * R] - BesselJ[m + 1, x * nd * R]) - nd * k * BesselJ[m, x * nd * R] *
        1 / 2 * k * n0 * (HankelH1[m - 1, k * n0 * R] - HankelH1[m + 1, k * n0 * R]), myPrec];
    Sqrt[Re[val]^2 + Im[val]^2]
    ];
(* supplementary functions *)
myFileName := "eigen." <> ToString[nInd[[1]]] <>
    "(" <> ToString[deltaN[[1]] <> ")_" <> ToString[nInd[[2]]] <> "(" <>
    ToString[deltaN[[2]]] <> ")_" <> ToString[R] <> "-M" <> ToString[mode];
plotResults[res_, box_, saveFormat_, useColor_] := Module[{plStyle, gr,
    fname, t = Table[{}, {4}], styles = Table[{}, {4}], modes, temp, k, color},
    If[res[[1]] == {}, Return[];];
    (*Off[Show::shx];*) (* hide warning for empty graph *)
    plStyle = {PlotRange -> All, PlotMarkers -> "●", Frame -> True,
      ImageSize -> {400, 300}, LabelStyle -> {FontSize -> 14}, Axes -> False};
    styles[[1]] = {plStyle, FrameLabel -> {"kR", "Re(k_mR)"}};
    styles[[2]] = {plStyle, FrameLabel -> {"kR", "Im(k_mR)"}};
    styles[[3]] = {plStyle, FrameLabel -> {"kR", "Q-factor"}};
    styles[[4]] = {plStyle, FrameLabel -> {"kR", "Absolute Value"}};
    If[! useColor,
```

```mathematica
        t[[1]] = ListPlot[Transpose[{res[[4]], res[[1]]}], styles[[1]]];
        t[[2]] = ListPlot[Transpose[{res[[4]], res[[2]]}], styles[[2]]];
        t[[3]] = ListLogPlot[Transpose[{res[[4]], res[[5]]}], styles[[3]]];
        t[[4]] = ListLogPlot[Transpose[{res[[4]], res[[3]]}], styles[[4]]];
      , (*else*)
        modes = Max[res[[6]]];
        Do[
            temp = Transpose[Select[Transpose[res], #[[6]] == k &]];
            color = ColorData["VisibleSpectrum"][380 + (750 - 380) * (k - 1) / modes];
            t[[1]] = Show[t[[1]], ListPlot[Transpose[{temp[[4]], temp[[1]]}],
        {styles[[1]], PlotStyle → color}], PlotRange → All];
            t[[2]] = Show[t[[2]], ListPlot[Transpose[{temp[[4]], temp[[2]]}],
        {styles[[2]], PlotStyle → color}], PlotRange → All];
            t[[3]] = Show[t[[3]], ListLogPlot[Transpose[{temp[[4]], temp[[5]]}],
        {styles[[3]], PlotStyle → color}],
      PlotRange → All, FrameTicks → {Automatic, Automatic}];
            t[[4]] = Show[t[[4]], ListLogPlot[Transpose[{temp[[4]], temp[[3]]}],
        {styles[[4]], PlotStyle → color}],
      PlotRange → All, FrameTicks → {Automatic, Automatic}];
          , {k, modes}];
    ];
    If[box || StringQ[saveFormat],
     gr = GraphicsGrid[{{t[[1]], t[[2]]}, {t[[3]], t[[4]]}}];,
     gr = GraphicsColumn[{t[[1]], t[[2]], t[[3]], t[[4]]}];
      ];
    If[StringQ[saveFormat],
     fname = myFileName <> "." <> saveFormat;
     Print["Saving graphics to " <> fname];
     Export[NotebookDirectory[] <> fname, gr]
     (* "Image" is not required and produces strange image file,
     but there is a bug in Mathematica 7 - it can not export ListLogPlot *)
     ];
    Print[gr];
  ];
(* saving data to the file *)
saveData[data_, lim_] := Module[{f, k, l, fname, res, uniqueVals, saveData, pos},
   If[data[[1]] == {}, Return[];];
   (* check same value *)
   uniqueVals = Union[data[[1]], SameTest → ((N[#1, lim] == N[#2, lim]) &)];
   If[Length[data[[1]]] ≠ Length[uniqueVals],
    Print["There are dublicated solutions. Try to perform more accurate calculations
        (starting points).\nThese values for Real part are:"];
```

```
          Print[Commonest[N[data[[1]], 12]]];
          Print["They will be removed from saved data"];
          saveData = Table[0, {Length[data]}, {Length[uniqueVals]}];
          For[k = 1, k ≤ Length[uniqueVals], k++,
            pos = Position[N[data[[1]], lim], N[uniqueVals[[k]], lim]];
            For[l = 1, l ≤ Length[data], l++,
              saveData[[l]][[k]] = data[[l]][[pos[[1]][[1]]]]];
            ];
            , (*if ... else*)
            saveData = data;
          ];
          (*res=Transpose[saveData];*)
          res = SortBy[Transpose[saveData], #[6] &];  (* sort by number
            of the level to manually select required levels from saved data *)
          fname = myFileName <> ".txt";
          f = OpenWrite[NotebookDirectory[] <> fname ];
          Do[
            WriteString[f,
              StringJoin[Riffle[ToString[FortranForm[N[#, lim]]] & /@ res[[i]], "\t"]] <> "\n"];
            , {i, 1, Length[res]}];
          Close[f];
          Print["Data saved to ", fname,
            " in the format Re(k_m R), Im(k_m R), Abs Value, k, Qfactor"];
        ];


  (* main part with the structure parameters *)
  myPrec := 50; (* working precision. should
    be larger than myGoals in approximately two times *)
  myGoals := 16; (* required precision for solution *)
  nInd := {1, 1.5}; (* n0 and nd *)
  R := 60;  (* radius *)
  stepsN := 15;  (* number of steps for changing index of refraction *)
  deltaN := {0, 0.00001}; (* change in complex part of the index of refraction *)
  mode := 130;  (* bessel's order *)


  (* calling parameters: k_min, k_max, k_points, km_min, km_max, km_points,
  number of low levels starting from km_min (set to -1 to show all levels within range) *)
  results = DiskModes[1.1, 2.1, 15, .7, 2.5, 100, 6];
  saveData[results, 16]; (* data, number of digits to save into file *)
  plotResults[results, True, "gif", False] (* data,
  2x2 (True) or 1x4 (False) output, Export graphics to "format"
    ("gif","jpg","eps".. check manual for Export supported formats) file (False to disable),
  use different color for each level (True) *)

  Do[
    mode := m;
    Print["Working with mode ", N[m], " ", DateString[]];
    results = DiskModes[.01, 1.5, 50, .01, 1.5, 50, 3];
    saveData[results, 16];
    plotResults[results, True, False, False];
    , {m, 55, 95, 10}]
```
Working with mode 5. Sun 30 Aug 2009 20:14:03

## Coupled disks:

```
n0 = 0; nd = 0; (* to make global, do not change it *)
DiskTE[mVal_, kmin_, kmax_, kpoints_, levels2find_] :=
   Module[{out = Table[{}, {7}], k, d, NN, same, levels0 = levels2find,
      kInt, xl, al, j, ReK, ImK, gImK, m, Nstep, startTime, endTime, g,
      countLevels, R = SetPrecision[R, myPrec], oldStart = {{}, {}}},
    Off[General::unfl];
    If[levels2find == -1, levels0 = 10^5;];
    xl = SetPrecision[Range[kmin, kmax, (kmax - kmin) / kpoints], myPrec];
    startTime = AbsoluteTime[];
    If[stepsN ≠ 0, Nstep = deltaN / stepsN, Nstep = {0, 0};];
    m = SetPrecision[mVal, myPrec];
    d = SetPrecision[dD, myPrec];
    NN = SetPrecision[Nval, myPrec];
    Do[ (*3*)
          countLevels = 0;
          Print["Working with k=", kInt, "  ", DateString[]];
         n0 = SetPrecision[nInd[[1]], myPrec];
         nd = SetPrecision[nInd[[2]], myPrec];
          k = SetPrecision[kInt, myPrec];
         al = fZeroImag[xl, m, k, R];
      Do[ (*2*)
           If[countLevels >= levels0,
            Break[];
            ];
            If[al[[g]] * al[[g + 1]] < 0,
            Do[ (*1*)
             If[j == 0,
               n0 = SetPrecision[nInd[[1]], myPrec];
               nd = SetPrecision[nInd[[2]], myPrec];
               ZeroImag = FindRoot[fZeroImag[x, m, k, R],
             {x, (xl[[g + 1]] + xl[[g]]) / 2}, AccuracyGoal → myGoals,
            PrecisionGoal → myGoals, WorkingPrecision → myPrec - 1];
```

```mathematica
    ReK = ZeroImag[[1]][[2]];

    ImK = -1 + signEq * 2 * BesselJ[2 * m, d * ReK * n0] * Cos[(k π)/(1 + NN)];


gImK = D[((-BesselJ[-1 + m, k0 * nd * R] + BesselJ[1 + m, k0 * nd * R]) * BesselY[m,
            k0 * n0 * R] + BesselJ[m, k0 * nd * R] * (BesselY[-1 + m, k0 * n0 * R] -
            BesselY[1 + m, k0 * n0 * R])) / (BesselJ[m, k0 * nd * R] * (-BesselJ[
                -1 + m, k0 * n0 * R] + BesselJ[1 + m, k0 * n0 * R]) + BesselJ[m, k0 *
            n0 * R] * (BesselJ[-1 + m, k0 * nd * R] - BesselJ[1 + m, k0 * nd * R])) +
        signEq * 2 * BesselY[2 * m, d * k0] * Cos[(k π)/(1 + NN)], k0] /. k0 → ReK;

    startRe = ReK;
    startIm = - ImK / gImK;

    ,

    If[Nstep[[1]] == 0 || deltaN[[1]] == 0,
       n0 = SetPrecision[nInd[[1]], myPrec];

       ,

       n0 = SetPrecision[nInd[[1]] + I * j * Nstep[[1]], myPrec];];
(* complex part of n0 *)
    If[Nstep[[2]] == 0 || deltaN[[2]] == 0,
       nd = SetPrecision[nInd[[2]], myPrec];

       ,

       nd = SetPrecision[nInd[[2]] + I * j * Nstep[[2]], myPrec];];
(* complex part of nd *)
       startRe = X[[1]];
       startIm = X[[2]];
      ];

    (*Print[{startRe,startIm}];*)
   value = FindMinimum[scatter[rek, imk, m, k, R], {rek, SetPrecision[startRe,
     myPrec]}, {imk, SetPrecision[startIm, myPrec]}, AccuracyGoal → myGoals,
 WorkingPrecision → myPrec, Method → "PrincipalAxis"];
  (*Print[value];*)
  X = {value[[2]][[1]][[2]], value[[2]][[2]][[2]]};
  Y = value[[1]] (* absolute function value *)
```

```
                   , {j, 0, stepsN}]; (*/1*)
        (* check whether these values were found before *)
         same = Length[Select[N[Transpose[out], 10],
            (#[[1]] == N[X[[1]] * R, 10] && #[[2]] == N[X[[2]] * R, 10] && #[[7]] == kInt) &]];
         If[same == 0, countLevels++, Print["Solution found before"]];


         out[[1]] = Append[out[[1]], X[[1]] * R];
         out[[2]] = Append[out[[2]], X[[2]] * R];
         out[[3]] = Append[out[[3]], Y];
         out[[4]] = Append[out[[4]], m];
         out[[5]] = Append[out[[5]], X[[1]] / Abs[X[[2]]]];
         out[[6]] = Append[out[[6]], countLevels];   (*number of the level*)
         out[[7]] = Append[out[[7]], kInt];
            ];

       , {g, 1, kpoints - 1}] (*/2*)

      , {kInt, 1, Nval}]; (*/3*)

    endTime = AbsoluteTime[];
    Print["Execution time:", (endTime - startTime), " seconds"];
    If[out[[1]] == {},
      Print["No solutions found, increase max K value or number of K points"];];
    out (* return array *)
    ];
fZeroImag[x_, m_, k_, R_] :=
   Module[{k0, kd, val, NN = SetPrecision[Nval, myPrec], d = SetPrecision[dD, myPrec]},
    k0 = x * n0;
    kd = x * nd;
    val = ((BesselJ[-1 + m, kd * R] - BesselJ[1 + m, kd * R]) * BesselY[m, k0 * R] +
          BesselJ[m, kd * R] * (-BesselY[-1 + m, k0 * R] + BesselY[1 + m, k0 * R])) /
       (BesselJ[m, kd * R] * (BesselJ[-1 + m, k0 * R] - BesselJ[1 + m, k0 * R]) +
          BesselJ[m, k0 * R] * (-BesselJ[-1 + m, kd * R] + BesselJ[1 + m, kd * R])) +
       signEq * 2 * BesselY[2 * m, d * k0] * Cos[(k π)/(1 + NN)];
    val
    ];
```

```
scatter[RK_, IK_, m0_, k_, R_] := Module[{k0 = SetPrecision[(RK + I * IK) * n0, myPrec],
    kd = SetPrecision[(RK + I * IK) * nd, myPrec], val, m = SetPrecision[m0, myPrec],
    d = SetPrecision[dD, myPrec], NN = SetPrecision[Nval, myPrec]},
  val = N[(n0 * HankelH1[m, k0 * R] * kd / 2 * (BesselJ[m - 1, kd * R] - BesselJ[m + 1, kd * R]) -
           nd * BesselJ[m, kd * R] * k0 / 2 *
            (HankelH1[m - 1, k0 * R] - HankelH1[m + 1, k0 * R])) /
         (nd * BesselJ[m, kd * R] * k0 / 2 * (BesselJ[m - 1, k0 * R] - BesselJ[m + 1, k0 * R]) -
           n0 * BesselJ[m, k0 * R] * kd / 2 * (BesselJ[m - 1, kd * R] - BesselJ[m + 1, kd * R])) +
       signEq * 2 * HankelH1[2 * m, k0 * d] * Cos[π * k / (1 + NN)], myPrec];
  Sqrt[Re[val]^2 + Im[val]^2]
  ];
(* supplementary functions *)
myFileName := "scat.general.sign" <> ToString[signEq] <>
    "." <> ToString[nInd[[1]]] <> "(" <> ToString[deltaN[[1]]] <> ")_" <>
    ToString[nInd[[2]]] <> "(" <> ToString[deltaN[[2]]] <> ")_" <> ToString[R];
plotResults[res_, box_, saveFormat_, useColor_] := Module[{plStyle, gr, fname,
    t = Table[{}, {4}], styles = Table[{}, {4}], modes, temp, k, color},
  If[res[[1]] == {}, Return[];];
  (*Off[Show::shx];*) (* hide warning for empty graph *)
  plStyle = {PlotRange → All, PlotMarkers → "●", Frame → True,
     ImageSize → {400, 300}, LabelStyle -> {FontSize → 14}, Axes → False};
  styles[[1]] = {plStyle, FrameLabel → {"k", "Re(kR)"}};
  styles[[2]] = {plStyle, FrameLabel → {"k", "Im(kR)"}};
  styles[[3]] = {plStyle, FrameLabel → {"k", "Q-factor"}};
  styles[[4]] = {plStyle, FrameLabel → {"k", "Absolute Value"}};
  If[! useColor,
       t[[1]] = ListPlot[Transpose[{res[[7]], res[[1]]}], styles[[1]]];
        t[[2]] = ListPlot[Transpose[{res[[7]], res[[2]]}], styles[[2]]];
        t[[3]] = ListLogPlot[Transpose[{res[[7]], res[[5]]}], styles[[3]]];
        t[[4]] = ListLogPlot[Transpose[{res[[7]], res[[3]]}], styles[[4]]];
    , (*else*)
       modes = Max[res[[6]]];
       Do[
            temp = Transpose[Select[Transpose[res], #[[6]] == k &]];
            color = ColorData["VisibleSpectrum"][380 + (750 - 380) * (k - 1) / modes];
            t[[1]] = Show[t[[1]], ListPlot[Transpose[{temp[[7]], temp[[1]]}],
```

```
                    {styles[[1]], PlotStyle → color}], PlotRange → All];
                        t[[2]] = Show[t[[2]], ListPlot[Transpose[{temp[[7]], temp[[2]]}],
                    {styles[[2]], PlotStyle → color}], PlotRange → All];
                        t[[3]] = Show[t[[3]], ListLogPlot[Transpose[{temp[[7]], temp[[5]]}],
                    {styles[[3]], PlotStyle → color}],
                PlotRange → All, FrameTicks → {Automatic, Automatic}];
                        t[[4]] = Show[t[[4]], ListLogPlot[Transpose[{temp[[7]], temp[[3]]}],
                    {styles[[4]], PlotStyle → color}],
                PlotRange → All, FrameTicks → {Automatic, Automatic}];
                      , {k, modes}];
        ];
        If[box || StringQ[saveFormat],
          gr = GraphicsGrid[{{t[[1]], t[[2]]}, {t[[3]], t[[4]]}}];,
          gr = GraphicsColumn[{t[[1]], t[[2]], t[[3]], t[[4]]}];
           ];
        If[StringQ[saveFormat],
          fname = myFileName <> "." <> saveFormat;
          Print["Saving graphics to " <> fname];
          Export[NotebookDirectory[] <> fname, gr]
          (* "Image" is not required and produces strange image file,
          but there is a bug in Mathematica 7 - it can not export ListLogPlot *)
          ];
        Print[gr];
      ];
(* saving data to the file *)
saveData[data_, lim_] := Module[{k, l, fname, res, uniqueVals, saveData, pos, f},
      If[data[[1]] == {}, Return[];];
      (* check same value *)
      uniqueVals = Union[data[[1]], SameTest → ((N[#1, lim] == N[#2, lim]) &)];
      If[Length[data[[1]]] ≠ Length[uniqueVals],
        Print["There are the same solutions. Try to perform more accurate calculations
             (starting points).\nThese values for Real part are:"];
        Print[Commonest[N[data[[1]], 12]]];
        Print["They will be removed from saved data"];
        saveData = Table[0, {Length[data]}, {Length[uniqueVals]}];
        For[k = 1, k ≤ Length[uniqueVals], k++,
```

```mathematica
      pos = Position[N[data[[l]], lim], N[uniqueVals[[k]], lim]];
       For[l = 1, l ≤ Length[data], l++,
         saveData[[l]][[k]] = data[[l]][[pos[[l]][[1]]]]];
      ];
       , (*if ... else*)
       saveData = data;
    ];
    res = Transpose[saveData];
    fname = myFileName <> ".txt";
    f = OpenWrite[NotebookDirectory[] <> fname ];
    Do[
      WriteString[f, StringJoin[
          Riffle[ToString[FortranForm[N[#, lim]]] & /@ res[[i]], "\t"]] <> "\n"];
       , {i, 1, Length[res]}];
    Close[f];
    Print["Data saved to ", fname,
      " in the format Re(kR), Im(kR), Abs Value, Mode, Qfactor, k"];
  ];


(* main part with the structure parameters *)
myPrec := 50; (* working precision. should
 be larger than myGoals in approximately two times *)
myGoals := 16; (* required precision for solution *)
nInd := {1, 1.5}; (* n0 and nd *)
R := 60;  (* radius *)
dD := (2 + 0.00001) * R; (* distance between disks *)
Nval := 2; (*number of disks*)
stepsN := 10;   (* number of steps for changing index of refraction *)
deltaN := {0, 0.00001}; (* change in complex part of the index of refraction *)
signEq := 1; (* defines sign in front of t in equation *)



(* calling parameters: m, k_min, k_max, k_points,
number of levels to find (-1 for 10^5 levels) *)
results = DiskTE[100, 0.5, 4, 200, 2];
saveData[results, 16]; (* data, number of digits to save into file *)
plotResults[results, True, "gif", False] (* data,
2x2 (True) or 1x4 (False) output, Export graphics to "format"
 ("gif","jpg","eps".. check manual for Export supported formats) file (False to disable),
use different color for each level (True) *)
```

```
AppendColumn[vector_List, matrix_List] := MapThread[Append, {matrix, vector}];
fData = {}; distances = {};
Do[
  dD := dist;
  results = DiskTE[100, 0.1, 4, 50, 1];
  fData = Append[fData, results];
  distances = Append[distances, dist];
  , {dist, 2.000001 * R, 2.1 * R, 0.01 * R}];
fData = AppendColumn[distances, fData];
saveData[fData, 16];
```