Monte Carlo Simulation Tool of Evanescent Waves Spectroscopy Fiber – Optic Probe for Medical Applications (FOPS 3D)

Daniel Khankin[1,2], Shlomo Mark[1,2] and Shaul Mordechai[3]
[1]Negev Monte Carlo Research Center (NMCRC), Sami Shamoon College of Engineering (SCE), Bialik/Basel Sts. Beer Sheva, 84100,
[2]Software Engineering Department, Sami Shamoon College of Engineering (SCE), Bialik/Basel Sts. Beer Sheva, 84100,
[3]Department of Physics, Ben-Gurion University, Beer-Sheva 84105,
Israel

1. Introduction

Fiber-optics evanescent wave spectroscopy has become a common technique for IR absorbance spectroscopy. Evanescent waves are formed when waves travelling in a medium undergo total internal reflection, which is the basis for the attenuated total reflectance (ATR) sampling technique [1,2]. Evanescent waves decrease exponentially as they propagates further away into the sample. Optical fibers which are transparent in the mid – IR are used as the ATR elements. The more reflections which cause the evanescent fields, the more efficient will be the measurement. Hence the curvature of the fiber and the radius of the uncladded part of the fiber, which brought in contact with the sample, play an important role in the efficiency of the measurement.

In this work, we describe a simulation tool (FOPS 3D) which can simulate full three dimensional geometrical structure of the fiber and the propagation of the light beam through the fiber. The simulation tool has been developed as a two iteration process of three steps each: alpha, beta, and release. The first iteration was a C# module (NMCRC-FEWs) [3] serving as an independent confirmation tool. After a "Software Test Plan", "Description", and "Report", the program moved (second iteration) to Java platform (FOPS 3D) and passed all the testing phases (requirements, design, program installation and regression testing). In addition, the FOPS 3D program favorably passed a black-box testing phase process of validation and verification.

 This application tool has the facility to freely bend the fiber, what gives the possibility to create any possible curvature folded fiber. Folding the fiber increases the number of beam hits in the uncladded part by slowing down the beam's propagation. A second feature of the tool is the possibility to change the radius of the uncladded part individually, specifically decreasing the radius relative to the fiber's radius, which increases the number of beam hits in the uncladded section. In addition, the fiber can be deformed and clay modeled, what gives the possibility to create different alternative geometrical shapes and inspecting their efficiency as ATR elements [2,3,4].

The simulation tool uses Monte Carlo methods for different parts of the simulation. There are two kinds of light emitters, an isotropic emitter and a Gaussian distribution oriented emitter. The simulation history is defined as a light beam, which hits the uncladded part and successfully travels through the fiber up to the light sensor. The number of successful histories is an estimator for the efficiency of the simulated fiber shape. The unbiased mean value of hits in successful histories is used for calculating the Fresnel transmission coefficients [1] being the absorption property of the simulated shape.

Thus, we first computed the relation between the radius of the uncladded part and the resultant absorption value. In order to find the most efficient geometrical shape, we then computed the relation between different folded fiber slopes with a constant radius of the uncladded section and the deduced absorption values.

The software engineering process for creating a simulation product is designed to systematically lead us from an initial conception of what the software is meant to do, to an implementation that correctly meets the objectives assigned to it [5].

When preparing or acquiring software for the simulation of fiber evanescent wave spectroscopy (FOPS 3D) processes, software engineering practices are invoked to force the software developer to recognize and confront issues before errors become embedded in a software product that does not work as a result. The preparation of software simulating a process begins with a problem to be solved, an environment representing the world in which the problem is to be formulated, a collection of stakeholders representing the diverse interests in the problem and successive reformulations of the problem as it moves from an imprecise "real world" form, to the resulting software product.

Developing a simulation tool invokes us to create a mathematical model for the process, a real – life or hypothetical situation in order to estimate its characteristics or solve problems in terms of the model. By changing states or variables of the model, we can predict the behavior of the system. With the advancement of computers and the improvement in ease of programming languages, computer simulation became a useful part of modeling systems, in variety of fields. In this article we present a simulation tool for optimizing fiber-optic probe structure for use in evanescent wave spectroscopy.

Evanescent wave spectroscopy is a very useful technique for inspecting and examining the properties of materials, mostly in the IR range. Due to the attenuated total reflection phenomenon, a ray which is completely reflected within the medium leaves an evanescent wave on the interface between the medium and an adjacent sample. The more reflections in the fiber, the more evanescent waves would be created, hence more signal will be absorbed. It is known that spectra of benign lesions and malignant cancerous regions have biochemical differences [2]. Thus, fiber evanescent wave spectroscopy is applied in biomedical skin diagnosis.

An important factor is the utilization of the spectroscopic method. Fiber wave spectroscopy is intensified if the absorbance of the fiber is increased. There are two ways to accomplish this. First it is possible to taper the untapped part of the fiber. It was investigated by [3] and by [4], who provided a correlation between the absorbance and the thickness of the untapped part. In other words, we would like to find the optimal width, d, of the untapped part.

Secondly, it's possible to bend the fiber about the untapped part. This action will make it harder for the light beam to progress and will make more hits in the untapped part, which will create more evanescent waves, thus increasing the absorption intensity. Briefly, we would like to find the optimal bending angle of the fiber with some constant width.
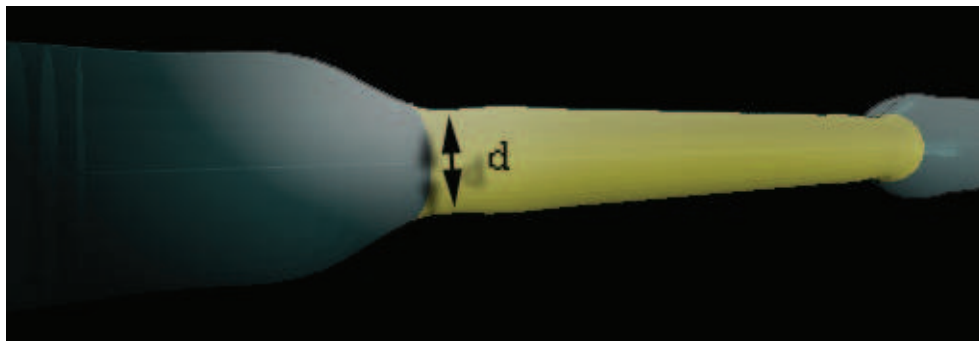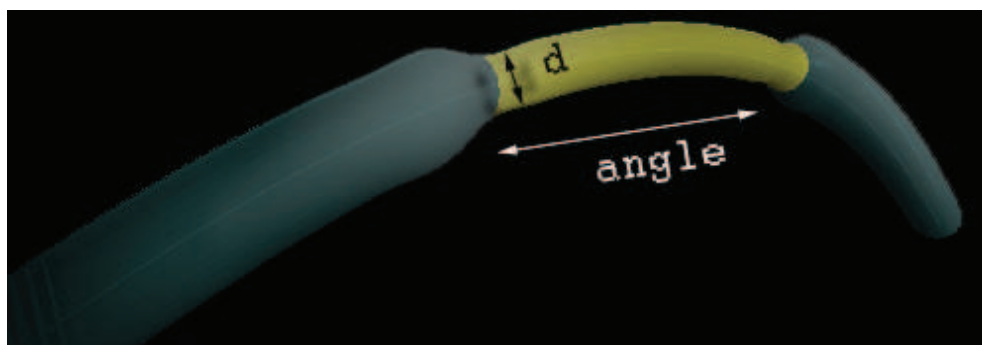
Fig. 1.

Fig. 2.

2. System description

The whole system consists of the simulated fiber, with properties such as the length, radius, reflection coefficients, and curvature radius of the midsection that can be set by user. There's an infrared simulated source, which emits rays according to a radial Gaussian distribution, uniform distribution and isotropous distribution.

The tool will simulate the advancement of the light beam through the fiber. The simulation will be based on Monte Carlo methods. Monte Carlo methods have been successfully applied in Chemistry, Physics, Medicine, or prediction of realistic behavior of an industrial system. The Monte Carlo methods approach, while simulating physical phenomena, is to create very large samples of random occurrences to recreate the dynamics of a system.

The purpose of the simulation is to measure the utilization of certain fiber geometry configuration. The simulation will estimate the probability of rays to successfully pass through the fiber in current geometry.

3. Physical background

The principle application of fiber evanescent wave spectroscopy in use is an IR source emits rays into a flexible optical fiber focused by a microscope objective. The radiation exits at the distal end of the fiber into an FTIR detector. The cladding of the middle section to be applied

to the region of interest is removed and immersed in a fluid. The taper is conical, narrowing down to a thinner cylindrical midsection, with index of refraction of the core n_1, the liquid index of refraction n_2, and n_{cl} of the cladding. To describe and sampling each incoming ray we need the radial distance r and angle θ relative to the z axis. When r is a Gaussian distribution for a beam waist given by [8]:

$$I(r) = I_0 e^{-8r^2/r_0^2} \tag{1}$$

r_0 is the radius of the laser aperture and I_0 is the irradiance at the center of the aperture.
The incidence angles of the rays must be less than $\theta_{cl} = \sin^{-1}(\sqrt{n_1^2 - n_{cl}^2})$ when θ_{cl} is the critical angle of the cladded section, and sampled from the angular distribution that given by [9]

$$P(\theta) \propto \begin{cases} \sin\theta / \cos^3\theta & \theta < \theta_{crl} \\ 0 & \theta \geq \theta_{crl} \end{cases} \tag{2}$$

According to the Beer-Lambert Law The transmitted power P(z) along a fiber is given by

$$P(z) = P_0 e^{-\gamma z} \tag{3}$$

Where P_0 is the power at the input face of the fiber, and γ is the effective evanescent absorption coefficient (varies for different orientations of the ray), which obey [10]:

$$\gamma(\theta) = \frac{\alpha\lambda n_2 \cos\theta \cot\theta}{2\pi r n_1^2 \cos^2\theta_c \sqrt{(\sin^2\theta - \sin^2\theta_c)}} \tag{4}$$

Where θ_c is the critical angle of the unclad section:

$$\theta_{cl} = \sin^{-1}\left(\sqrt{n_1^2 - n_{cl}^2}\right) \tag{5}$$

And α is the bulk specific absorption coefficient of the cladding material at the given wavelength λ [11]

$$\alpha = A e^{-\gamma v/v_0} \tag{6}$$

v is the optical wavenumber (1/λ) and A, γ, and v_0 are constants given in data tables [12].
For a simple tube of constant radius, $\gamma(\theta)$ can be computed by integration of Equation 4 over the power distribution of Equation 2. This is very difficult and inapplicable in a condition where the incidence angles change at each collision with the interfacing surface, especially when we use a non-point source emitted rays. Thus, in our Monte Carlo approach we have used Ruddy's equation [10]

$$\gamma = NT \tag{7}$$

where N is the number of reflections per unit length and T is the Fresnel transmission coefficient for an evanescent wave, which, for any angle θ_z relative to the core axis, is:

$$T(\theta_z) = \frac{\alpha\lambda n_2 \sin\theta_z}{\pi n_1^2 \sin^2\theta_c \sqrt{(\cos^2\theta_z - \cos^2\theta_c)}} \tag{8}$$

Where,

$$\theta_c = \cos^{-1}(n_2 / n_1)$$

is the complementary critical angle.

These formulas apply only to rays above the critical angle. The conical section, however, reflects many incident rays below the critical angle. In these cases, the ray is quickly attenuated completely, as the Fresnel power coefficient for rays not completely reflected is:

$$\frac{I_r}{I_0} = \frac{(n_1 \cos\theta_1 - n_2 \cos\theta_2)^2}{(n_1 \cos\theta_1 + n_2 \cos\theta_2)^2} \tag{8}$$
$$+ \frac{(n_2 \cos\theta_1 - n_1 \cos\theta_2)^2}{(n_1 \cos\theta_2 + n_2 \cos\theta_1)^2}$$

Where, I_r is the intensity of the reflected ray and I_0 of the incident ray.

In the Monte Carlo simulation, rays are fired into the fiber according to the radial and angular distributions of Equations 1 and 2, respectively.

The ray movement is defined by five variables: Three Cartesian coordinate – to define the particle location, and two angles defining the direction of movement, or alternatively using three Cartesian coordinate and the projection on each axis. In order to define the new location we define three variables:

1. Location, defined by the vector $\vec{r}$, when $\vec{r} = (x_{old}, y_{old}, z_{old})$

2. Direction of movement $\vec{u}$, when $\vec{u} = (u_{in}, v_{in}, w_{in}) = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta)$ where the angles θ, ϕ defined relative to the Z.

3. Distance to the geometric boundary ds.

Thus, $\vec{r} = \vec{r} + \vec{u} \cdot ds$, when $\vec{r} = (x_{new}, y_{new}, z_{new})$, therefore

$$x_{new} = x_{old} + u_{in} \cdot ds$$
$$y_{new} = y_{old} + v_{in} \cdot ds$$
$$z_{new} = z_{old} + w_{in} \cdot ds$$

At the geometric boundary we get changes in the direction, and the new directions are defined by new scattering angles Θ, Φ .

In order to define the new direction we have to use the operator

$$\Re = \begin{pmatrix} \cos\theta\cos\phi & -\sin\phi & \sin\theta\cos\phi \\ \cos\theta\sin\phi & \cos\phi & \sin\theta\sin\phi \\ -\sin\theta & 0 & \cos\theta \end{pmatrix}$$

Therefore the new projections are

$$(u_{new}, v_{new}, w_{new})$$

$$u_{new} = u_{old} \cdot \cos\Theta + \sin\Theta \cdot (w_{old} \cdot \cos\Phi\cos\phi - \sin\Phi\sin\phi)$$
$$v_{new} = w_{old} \cdot \cos\Theta + \sin\Theta \cdot (w_{old} \cdot \cos\Phi\sin\phi + \sin\Phi cos\phi)$$
$$w_{new} = v_{old} \cdot \cos\Theta - \sin\Theta \cdot (\sin\theta \cdot \cos\Phi)$$

The simulation process is based on realization of the beam movement in a three dimensional virtual fiber. We outline the Monte Carlo approach to fiber evanescent wave spectroscopy by firing rays according to a given distribution into an optical fiber with a ROI (the unclad section) and ended with a detector. In the Monte Carlo tool we modeled the absorbance at each point of reflection, and the net absorption of the ROI - if θ is above the critical angle of the unclad section, the ray is discarded; otherwise, the number of collisions encountered within the unclad section are counted and multiplied by γ/α to provide its absorbance "weight". This is averaged over a large amount of rays fired (histories) into the fiber to yield the average relative absorbance.

4. Development process

The development process is a very important aspect, especially in heavy computational applications. In addition to the common problems inherent in software development, this simulation tool goes under the category of scientific software. Scientific software is addressed towards scientific computations [7]. The purpose of which is usually some reconstruction or prediction of phenomena or process from science. It is also another way to study a phenomena besides theory and experiment. Scientific software is susceptible to complex scientific loads. Furthermore, it requires a knowledgeable user, who should know how to use and operate the software. Consequently, he should also understand the input for the software and the output. One more important requirement is a knowledgeable tester. Person, who is responsible for testing the software, obviously should also be aware to the mathematical models being used. The software engineering process for creating a simulation product is meant to systematically lead us from an initial conception of what the software is meant to do, to an implementation that correctly meets the objectives assigned to it [5]. In the initial step of the software engineering process, project initiation document, difficulties could be seen. It is very unclear what exactly has to be done, and how the simulation tool has to work. Proceeding to the software requirement specifications document (SRS) didn't solve the issues. It's very unclear what exactly the client wants, for the reason that the user has strong knowledge in physics, usually, as opposed to the developer. The developer may understand what the client says, and understand the formulae but he doesn't grasp their exact meaning, and the conclusions they predict. On the other hand, the client is not thoroughly knowledgeable about constraints and difficulties the developer has. Some of them are very significant, such as floating point precision, round off errors, parallel computations, running time, and complexity of implementing algorithm. They may constitute to correct working of application or vice versa be the reason for failure. It's like a big gap between the developer and the client, two separated worlds. One understands the words of the other but doesn't understand their supposed meaning. Hence, different approach must be taken. Specifically, the Agile software development methodology was chosen to be the basis for working on the project. Short on words, agile methodology governs important rules. The whole task must be broken into small units with minimal planning. Each unit development involves full software development cycle. Contact with the client on daily basis, preferably face to face over documents. Collaborative work with the customer (doesn't has to be the client). Secondly, test driven development methodology should be incorporated (TDD). TDD states, that the test cases should be created first, and only afterwards codes the functionality which will pass the test case. Code requirements must be defined before writing the code. The tests will contain assertions that either true or false. Since the whole work was broken into small units, Black Box testing was easily adapted.

Taking an external perspective of the test object, which is one unit of work, to derive test cases. Tests are usually functional, treating the unit as black box with no knowledge of internal structure. Tests determine correct output on valid or invalid input. FOPS3D was developed on Java, for portability of the application and independence of floating point calculations in hardware. Furthermore, for the developer to waste less time on memory managements and generic algorithms development, what could be in the case of using C++. The software was developed under the requirements and conditions of Object Oriented Analysis Design and Programming with particular attention to practical techniques such as UML (Unified Modeling Language) diagramming and design patterns. The use of iterative design, parallel development and high modularity all enhance maintain ability, ease of verification and code reusability. To ensure a high quality program and improve reliability, FOPS3D has passed all the software testing and verification phases and methods – requirement phase testing, design phase testing, program phase testing, automated black box testing, walkthrough, and white box testing. The last verification came when the results showed agreement with previous experimental and with the analytical calculations. After we achieved those agreements we could say that FOPS3D present a powerful tool for analyzing Fiber-Optics probe, where detailed experiments are too cumbersome to undertake.

5. Development characterization

The simulation tool was developed in Java language, utilizing object oriented and event driven concepts of the language. Furthermore, with the help of Java standard API, it was possible to concentrate on the mathematical model and development of the simulation tool itself using the native abstract data structures and collections which are part of the Java platform. The graphics was programmed using OpenGL API, without decreasing the portability of Java. The computations of the fiber's surfaces and geometrical structure, and of the ray beam advancement and collision, were distributed among multiple threads, in order to achieve maximal CPU utilization.

6. Application structure

The program is categorized into logical packages subdivision. There are a total of five packages:

Package Name	Package Description
common	Contains objects, data structures and interfaces used through the application.
forms	Contains GUI elements.
graphics	Contains graphic objects, and the rendering engine.
math	Contains the mathematical model for fiber construction, mathematical
root	Application loader.

The fiber entity is represented with the help of two data structures contained in two classes. The 'Fiber' class and the 'FiberPart' class which composes the Fiber. The 'FiberPart' class represents the smallest section of the fiber. There, all fiber parts are connected among

themselves to construct the whole fiber. In order to implement the geometrical elasticity effect, the fiber was uniquely designed. Among the central axis, control points were positioned, and an interpolation was made to pass a curve through those control points. The fiber surface is built around this curve. This gives the possibility to bend the fiber. As soon as the user moves one of the control points, the curve is re-interpolated and the fiber's surface is rebuilt. On the fiber's surface different control points are placed. With the same principle, another curve is passed through those control points. This curve sets every fiber part's shape, by determining the radius of every fiber part independently. By moving the exterior control points, the radius of the fiber parts is altered appropriately, what gives the possibility to freely model the fiber's geometrical structure.

The ray is implemented by the 'Ray' class. The ray class samples the initial emission angles, and performs the computation for ray movement and collisions.

During the simulation process, no graphics are displayed in the sake of centralizing all CPU time on simulation computations.

Those are the main classes of the simulation application:

Fiber

Fiber class which is responsible for the fiber structure, and holds all the data structures which combine the fiber. In addition, the visual representation of the fiber is implemented in this class.

FiberPart

It's a data structure which hierarchically combines the fiber, and holds the normal vectors to the fiber's surface. This data structure plays an essential part in the simulation's computations.

Ray

Class which holds the entire ray beam data regarding of its spatial position points and its visual representation.

RayWorker

Worker class which is responsible for the calculation of the advancement and reflections of a ray beam. For each ray the calculations is performed on different thread.

FiberSurfaceWorker

Worker class which is responsible for the mathematical model of the fiber, the calculation and recalculation of fiber surface and shape as the user performs deformation of the fiber's surface and bends it. The calculation of the fiber's shape is done on different thread.

SimulationEngine

The engine of the simulation application, the engine connects all different parts of the simulation and responsible for the whole procedure of the simulation.

The visual and the computational modules are isolated from each other and controlled by the simulation unit.

7. Simulation results

The simulation ran for different tapered midsection radii and different bending angles while the midsection's radius is constant. For every different setting of fiber, the simulation ran for 1,000,000 histories or for $\rho \leq 0.1$. The midsection type for both simulations was radial and the results were compared with the results from [3].

The first results are when the fiber is not bent, and the simulation ran for different midsection radii while the radius of the fiber is constant.

Those results are in accordance with those presented in [3], where tapped midsection was used. It's possible to see that the results are slightly different, because of the different
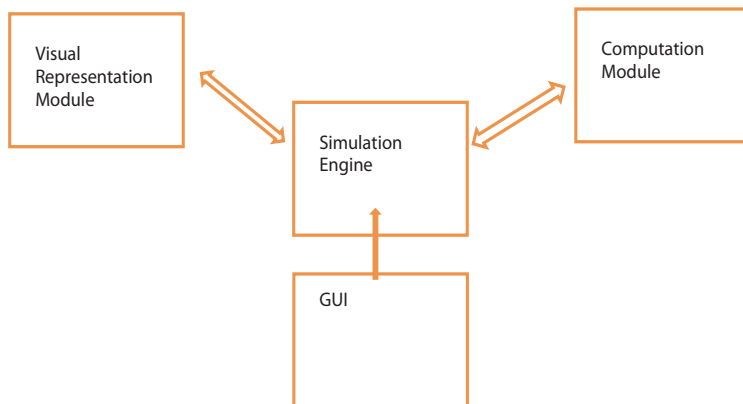
Fig. 3.

midsections used in the simulation. Thus, our results clearly indicate that using thinner tapered midsection radii may increase the absorbance intensity dramatically and improve spectral signal to noise ratio.

Property	Value
Fiber radius	0.9mm
Midsection radius	0.2mm – 0.36mm
n_1	1.457
n_2	1.357
Fiber length	10cm
Midsection length	5cm
Histories	1,000,000

The results are shown in Fig. 4.
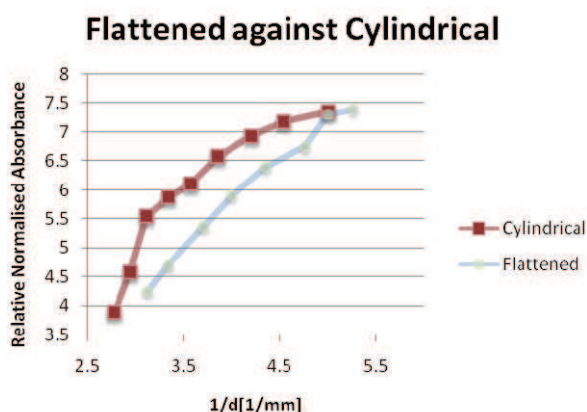
Fig. 4.

Program user manual

1. Popup menu

Right clicking the mouse bring a popup menu, in which the user can select one of the three modes: View Perspective Change, Bend Fiber, Deform Fiber.
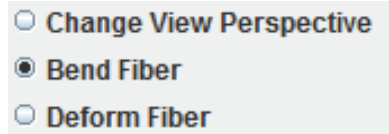
1.1 Perspective Change

The fiber being simulated exists in a 3D world, in a right-handed coordinate system, i.e. the negative Z axis points into the screen, the positive Y is pointing up and the positive X axis is pointing right. The user can rotate the viewing position about the X and Y axis and zoom in/out.

1.1.1 Viewpoint rotation

while holding the left mouse button and moving the mouse left or right will rotate the viewing position about the Y-axis accordingly. Moving the mouse up or down will rotate the viewing position about the X axis accordingly.

1.1.2 Zoom In/Out

Using the mouse wheel, the user can zoom in or out. Rotating the wheel forward will zoom in and rotating the wheel backward will zoom out.

1.1.3 Fiber bending

For bending the fiber the user should move the big control points. When the user moves the mouse over one of those control points they will highlight for letting know that they can now be moved. When the control point is highlighted, hold left mouse button to grasp the control point. Moving the mouse will also move the control point causing the fiber to bend itself. There are total four control points for bending the fiber. One at each end of the fiber, and two among the body of the fiber.
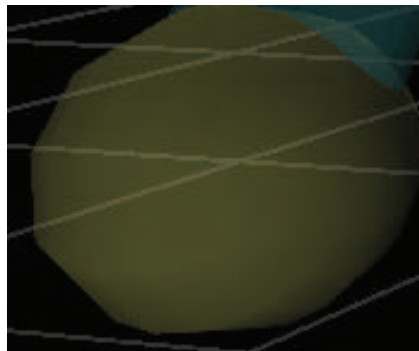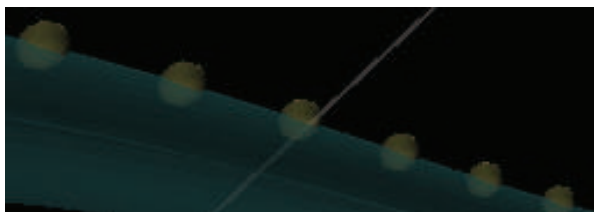

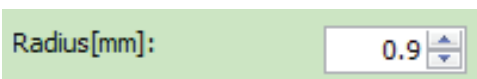It's possible to bend the fiber from a spinner component and setting the bending angle.

1.1.4 Fiber deformation

For deforming the fiber the user should move the small control points. When the user moves the mouse over the small control point it will highlight for letting know that they can now be moved. When the control point is highlighted, hold left mouse button to grasp the control point. Moving the mouse will also move the control point cause the fiber to deform its shape. The control points are among the top edge of the fiber.
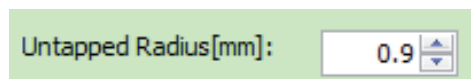

1.1.5 Fiber and/or midsection radius

The fiber's radius can be set using the spinner component.

For midsection part, it's possible to change its radius from a spinner component.

### 1.2 Simulation properties

1.2.1 Fiber properties

Fiber properties of the simulated fiber which can be changed are:
the refractive index of the cylindrical core n_1, the refractive index of the cladding n_2, the fiber's inner radius, fiber length, untapped part length, untapped part radius.

1.2.2 Simulation properties

The number of iterations (histories) can be changed and its number would be accounted in the overall calculation.

### 1.2.3 Midsection type

From menu, positioned at the bottom, it's possible to choose whether the midsection is radial or flattened.
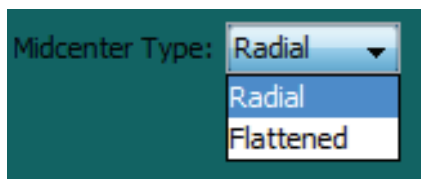

1.3 Selecting untapped part

For the simulation to begin, the untapped part must be set. This can be done in either the fiber bending mode or the fiber deformation mode. While in one of those modes,

select a small control point by left clicking with the mouse. Its color will be changed to notify that it's selected. Clicking the right mouse button will bring a popup menu where the user can set whether the selected control point is the left end of the untapped part either the right end.
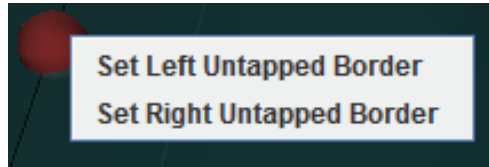
After setting both left and right ends, the untapped part of the fiber will be marked in different color. Afterwards, the simulation can be started.
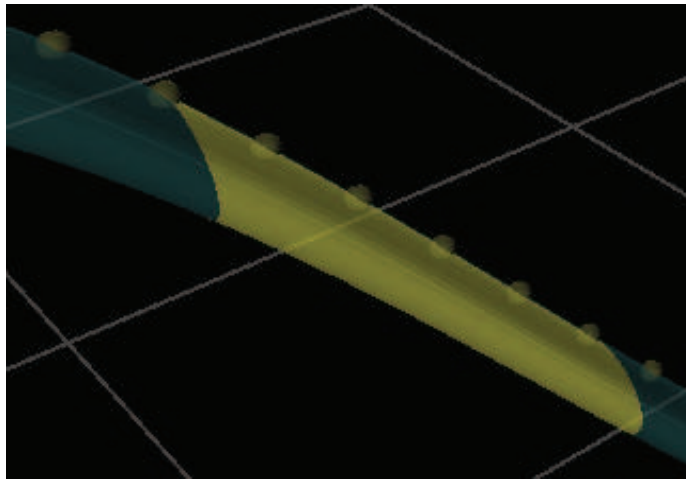
1.4 Starting simulation

Clicking the start button ![start button] will start the simulation by sending beams of light, which advancement can be tracked in two types of animation window, 3rd person view and 1st person view. The animation can be turned off to offload the CPU. Before starting the simulation, it's necessary to set the number of iterations.

1.5 Pausing the simulation

The simulation can be paused at any moment by clicking the pause button

, and clicking it again will resume.

1.6 Managing rays

On the left part of the screen, a table is shown with every ray which is being in the fiber at the moment and its advancement progress.

ID	Status	Progress
1		59%
2		71%
3		43%
4		91%
5		42%

8. Reference

A. Messica, A. Greenstein, and A. Katzir, *Theory of fiber – optic evanescent wave spectroscopy and sensors. Applied Optics*, Vol. 35, Issue 13, pp. 2274-2284 (1996).

Y. Raichlin, L. Fel, and A. Katzir. *Evanescent-wave infrared spectroscopy with flattened fibers as sensing elements*. Optics Letters, Vol. 28, Issue 23, pp. 2297-2299 (2003)

Mann MP, Mark S, Raichlin Y, Katzir A and Mordechai S., *Optimization of fiber-optic evanescent wave spectroscopy: a Monte Carlo approach*. Appl Spectrosc. 2009 Sep;63(9):1057-61.

B. D. Gupta, C.D. Singh, A. Sharma, *Fiber optic evanescent field absorption sensor: effect of launching condition and the geometry of the sensing region*, Optical Engineering, Volume 33, No. 6, pgs. 1864-1868, June (1994)

Sommerville, I, Software engineering, 7th edition, New York: Addison Wesley, (2004)

A Dubi, *Monte Carlo Applications in Systems Engineering*: John Wiley & Sons, 2000.

Daniel Khankin, Alan D Solomon, Yosef Shpungin, Yoram Shtoland, and Shlomo Mark, "A Monte Carlo Package for Optimizing Fiber - Optic Evanescent Wave Spectroscopy as a Test Case for Scientific Programming" in *SMRLO*, Beer Sheva, Israel, 2010.

E. Hecht, *Optics*, Addison Wesley 2002

A. W. Snyder and J. D. Love, Optical Waveguide Theory (Chapman and Hall, London, 1983).

V. Ruddy, B. D. MacCraith, and J. A. Murphy, J. Appl. Phys. 67, 10 (1990).

T. F. Deutsch, J. Phys. Chem. Solids 34, 2091 (1973).

R. Waynant and M. Ediger, Electro-Optics Handbook (McGraw Hill Professional, New York, 2000), 2nd ed.

Applications of Monte Carlo Methods in Biology, Medicine and Other Fields of Science

Edited by Prof. Charles J. Mode

ISBN 978-953-307-427-6

Hard cover, 424 pages

Publisher InTech

Published online 28, February, 2011

Published in print edition February, 2011

This volume is an eclectic mix of applications of Monte Carlo methods in many fields of research should not be surprising, because of the ubiquitous use of these methods in many fields of human endeavor. In an attempt to focus attention on a manageable set of applications, the main thrust of this book is to emphasize applications of Monte Carlo simulation methods in biology and medicine.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Daniel Khankin, Shlomo Mark and Shaul Mordechai (2011). Monte Carlo Simulation Tool of Evanescent Waves Spectroscopy Fiber – Optic Probe for Medical Applications (FOPS 3D), Applications of Monte Carlo Methods in Biology, Medicine and Other Fields of Science, Prof. Charles J. Mode (Ed.), ISBN: 978-953-307-427-6, InTech, Available from: http://www.intechopen.com/books/applications-of-monte-carlo-methods-in-biology-medicine-and-other-fields-of-science/monte-carlo-simulation-tool-of-evanescent-waves-spectroscopy-fiber-optic-probe-for-medical-applicati

INTECH
open science | open minds