LIGHT TRANSPORT IN BIOLOGICAL TISSUES USING GPU

TRANSPORTE DE LUZ EN TEJIDOS BIOLÓGICOS UTILIZANDO PROCESADORES GRÁFICOS

N. López-Marín[a†], F. Corrales-Machín[b], E. Castro-Alvarez[c] y R. Rodríguez- Fernández[a,c]

a) Facultad de Física, Universidad de La Habana, Cuba; nlopez@fisica.cu †
b) Instituto de Geofísica y Astronomía, La Habana, Cuba
c) Instituto de Nefrología, La Habana, Cuba
† corresponding author

Recibido 4/7/2016; Aceptado 10/11/2016

The exact solution of the propagation of light in turbid media is possible only in very simple problems. In almost all practical cases numerical methods are mandatory. In this paper we calculate the absorbance of light in tumoral tissue using Monte Carlo (MC) simulation in order to optimize the execution time of several parallel algorithms for Graphic Processing Units (GPU) and a serial code running on the Central Processing Unit (CPU) for 10^6 to 10^{10} photon packets. The plots of absorbance versus time and tissue depth are presented, showing that the precision of the methods depend on the number of photons and it is algorithm independent. The implementation of MC algorithms using GPU have shown that simulations may be 300 times faster than on a CPU providing an effective time framework to study complex systems.

La interacción de la luz con medios turbios complejos se estudia empleando métodos numéricos. En este trabajo presentamos un modelo para el cálculo de la absorbancia de la luz en tejido tumoral utilizando el método de Monte Carlo (MC) con el objetivo de optimizar los tiempos de ejecución de varios algoritmos paralelos ejecutados en unidades de procesamiento gráfico (GPU) y un código serie en la unidad central de procesamiento, variando el número de paquetes de fotones desde 10^6 hasta 10^{10}. Presentamos los gráficos de absorbancia en función del tiempo y la profundidad del tejido demostrando que la presición del método aumenta con el número de fotones y es independiente del algoritmo utilizado. Demostramos que el uso de GPU puede aumentar la velocidad del método 300 veces, siendo una solución para estudiar este tipo de problemas.

PACS: Monte Carlo simulations 87.10.Rt, Light absorption and transmission 42.25.Bs, Photodynamic therapy, 87.50.wp

I. INTRODUCTION

Photodynamic therapy(PDT) is a minimally invasive therapeutic procedure used very often in the treatment of skin cancer [1]. It includes the uptake of a photosensitive compound in the tumor and the local activation of the drug by delivering a light dose at a specific wavelength onto the region of interest. In order to obtain a good outcome, accurate light dosimetry is critical during the treatment. Among other techniques for computing light dose, Monte Carlo (MC) methods are the most used in terms of accuracy and flexibility, been able to score multiple physical quantities simultaneously [2]. The Monte Carlo method is a statistical one, relying on the calculation of the propagation of a large number of photon packets. Consequently, it requires a large amount of computational time. Several techniques have been developed to speed up the simulation, among them the implementation of Monte Carlo algorithms using Graphic Processing Units [2].

In this paper we present a model for the propagation of light in a tumor in order to optimize the execution time of several algorithms running in the Graphic Processing Unit (GPU) and compare them with a serial algorithm running in the Central Processing Unit (CPU). In the first section we present the relevant theoretical aspects of light propagation in tissue and a description of the model we used. Then, we describe the implemented algorithms and the computer system used to execute them. Finally, we present the results of the absorption probability density matrix, the execution time optimization and the conclusions of our work.

II. LIGHT PROPAGATION IN TISSUES

The propagation of light in heterogeneous materials is determined by the Radiation Transport Equation (RTE) [3]. But, for almost all cases of practical interest, an analytic solution of the RTE is not possible and the use of approximate methods is mandatory. The simplest and most widely used approach is to replace the RTE with a diffusion equation for the fluence rate [4]. This approximation assumes that the radiance is isotropic and requires that the point of interest is far from the source of light and the boundaries of the system. Unfortunately, these assumptions are often unrealistic for PDT protocols because biological tissues scatters light not isotropically, mainly in the forward direction.

A general approach to solve the RTE with all its complexities is the Monte Carlo method [2]. An accurate description of the technique can be found in the literature [5–7]. Figure 1 shows a diagram of the algorithm used for the Monte Carlo method. In this model a photon packet with a given weight w, proportional to its energy, is launched into the tissue. At each time step, the packet moves with a step size defined by equation 1, where s is the step size, ξ is a random number uniformly distributed between 0 and 1, μ_a the absorption coefficient and μ_s the scattering coefficient.

$$s = -\frac{\ln(\xi)}{\mu_a + \mu_s} \tag{1}$$

At every interaction site, the weight of the packet w is reduced by Δw due to the tissue absorption (Equation 2). This value is

also accumulated in the absorption matrix at the grid point to record the amount of energy deposited at every interaction point.

$$\triangle w = w \frac{\mu_a}{\mu_a + \mu_s} \tag{2}$$

After been absorbed, the photon packet changes its direction due to the scattering process. The new direction is characterized by the azimuthal and deflection angles, Φ and Θ respectively. Φ is uniformly distributed between 0 and 2π. The probability distribution function of the cosine of the deflection angle is determined using the Henyey-Greenstein function [8], equation 3, where g is the anisotropy factor and equals $< cos(\Theta) >$.

$$P\left(cos(\Theta)\right) = \frac{1}{2} \frac{1 - g^2}{(1 + g^2 - 2g\cos\Theta)^{\frac{3}{2}}} \tag{3}$$

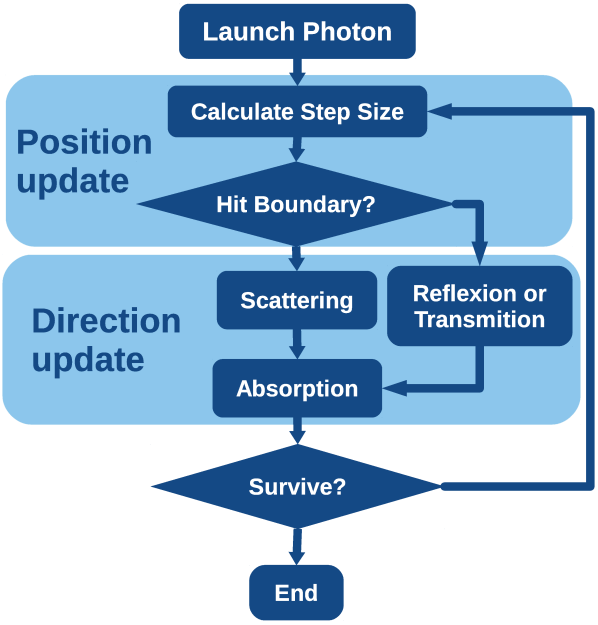

Figure 1. Diagram of the algorithm for Monte Carlo simulation of the light absorption in tumor tissue.

When the photon packet hit a boundary is either reflected or transmitted. The step size is reduced to the point in which it hits the boundary. We determine whether the photon is internally reflected or transmitted by calculating the internal reflectance, r, using Fresnel's formulas [9] and a random number uniformly distributed, ξ. If $\xi \leq r$, then the packet is internally reflected; if $\xi > r$, it is transmitted. The new directions are calculated by Snell's law in case of transmission and by reverting the directional cosine of the component perpendicular to the boundary in case of reflexion.

The process is repeated until the photon packet leaves the tissue or its weight fall bellow 90 %. In the former case a Russian roulette gives the photon packet one chance out of ten of surviving with their weight multiplied by 10 in order to ensure energy conservation. When a sufficient number of packets are launched, the cumulative distribution of all photon paths provide an accurate approximation to the true solution of the light transport problem.

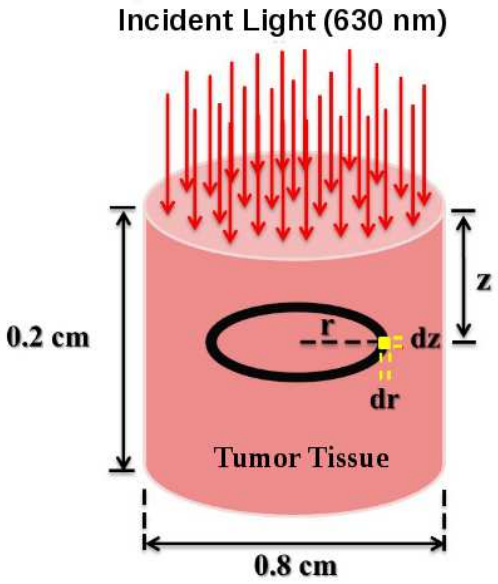

Figure 2. Representation of the tumor tissue layer with $0.2cm$ of depth and $0.4cm$ of radius, illuminated with a light beam at $630nm$ perpendicular to the surface.

In our model the tumor is simulated by a cylinder of $0.4\,mm$ of radius and $0.2\,mm$ of depth with optical parameters $\mu_a = 1.7\,cm^{-1}$, $\mu_s = 125\,cm^{-1}$, $g = 0.8$ and refraction index, $n = 1.37$, taken from ref. [10] for basal cell carcinoma at $630\,nm$. At this wavelength the penetration of light is deeper in biological tissues. During the simulation the absorption weight is scored in a two dimensional array in which each element $A[r, z]$ represents the cumulative photon weight in a ring of radius r, located at a depth z with thickness $dr = 0.0005\,cm$ and height $dz = 0.0005\,cm$. The absorbed photon probability density is calculated dividing the cumulative photon weight at each grid point by the number of photon packets and the volume of the region, in our case $\pi dr(2r + dr)dz$.

The tissue is illuminated with a light beam perpendicular to the surface. In order to simulate this process, the initial x and y coordinates of the photon packet are randomly generated in a circle of $0.4\,cm$ of radius and the z coordinate is set to zero. Figure 2 shows a representation of the model. In our model, we do not consider refraction or reflexion because we have the same optical properties in all the tumor tissue.

III. ALGORITHMS IMPLEMENTATION

The main limitation of Monte Carlo method for obtaining a RTE numerical solution is the requirement of extensive computations. Fortunately, in the last few years several techniques have been developed to speed up and simplify the simulations [5, 11]. Graphic cards have evolved into multiprocessors with high computational power do to the heavy calculations needed for the high definition, real time 3D graphics. The development of nVidia GPU technology

and the CUDA (Compute Unified Device Architecture), an extension to the C programming language, made possible the spread of the use of graphic cards exclusively for calculations [12]. In the CUDA extension, the program routine written for running in the GPU and called by the serial part is named kernel. The GPU itself is composed of several Streaming Multiprocessors (SMP) each one containing many CUDA processing cores managed by a common control unit. The blocks running on the SMPs manage the parallel threads running on the CUDA cores.

The implementation of Monte Carlo algorithms using Graphic Processing Units (GPU), has shown that simulations may be many times faster than on a single standard Central Processing Unit (CPU) [13] providing an effective framework to study complex systems.

A serial program, written in ANSI C language, was made with the algorithm proposed in the previous section. We call it S02. In order to obtain the optimal parallel program we implemented several codes using the algorithms described in Figure 1. In all of them the Marsaglia XORWOW algorithm [14] was used to generate the uniformly distributed random numbers. MP^2 blocks and 512 threads were used, where MP represents the number of multiprocessors read in the programs from the graphic card. We wrote the absorption matrix using atomic operations to avoid a race condition if several threads attempt to write on the same memory address at the same time.

Table 1. Description of the implemented codes.

Code	Description
S02	Serial Code
A01	The kernel calculates one photon packet each time it is called. The iteration occurred in the CPU
A02	Each kernel is called only once calculating many photon packets. The iteration packets occurred in the kernel
A06	Single precision fast math operations of the CUDA core Arithmetic Logic Unit (ALU) is used
A09	Two graphic processors installed on the computer, fast math, intrinsic functions and pinned memory, allowing the GPU to manage the host address space.

Table 1 shows the main characteristics of each program code. On the A01 code, the kernel calculates only one photon packet at a time and the CPU code iterates until the total number of photon packets, N, are calculated. The iteration occurred in the CPU. On the A02 code, N is divided in such a way that each kernel is called only once calculating many photon packets. The iteration over multiple photon packets occurred in the kernel. A06 code is like A02, but using single precision fast math operations of the CUDA core Arithmetic Logic Unit (ALU). In A09 all possible optimizations for the GPU code are used: two graphic processors installed on the computer, fast math, intrinsic functions and pinned memory, allowing the GPU to manage the host address space.

All the programs were executed changing the amount of photon packets from 10^6 to 10^{10}. The simulations were performed on a computer with 2 AMD Opteron processors of the 6800 series with 12 cores each, 128 GB of RAM memory, a hard disk array of 3 TB of capacity and 10000 rpm and 2 nVIDIA Tesla C2050 graphic cards with 448 processors and 4 GB of GDDR5 graphic memory.

IV. ABSORPTION PROBABILITY MATRIX

When two Monte Carlo simulations, using the same parameters are performed, there are differences in the results produced by the statistical variability of the stochastic method.

In order to show that all the programs are equivalent in terms of the results obtained, we compared the difference between two simulations running the same code and two different codes to show that the variability is the same.

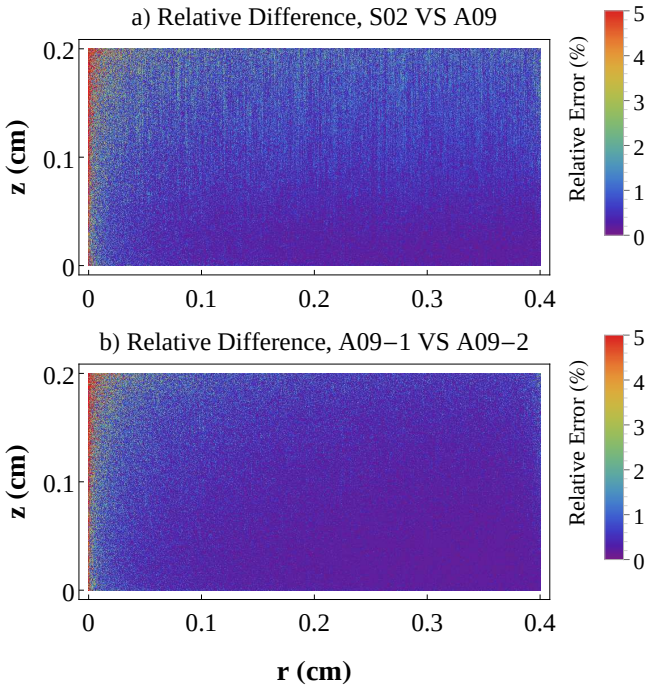

Figure 3. Relative difference in the absorption photon probability when comparing two simulations. (a) Relative difference between the simulation S02 and A09 using 10^9 photon packets. (b) Relative difference between two simulations with A09 using 10^9 photon packets. In both cases the difference is smaller than 5 % demonstrating that the serial and parallel codes are equivalent in terms of results obtained.

The relative difference in absorption probability, E_{ij}, is calculated using equation 4. Where $A1_{ij}$ and $A2_{ij}$ represent the matrices we wanted to compare.

$$E_{ij} = \frac{|A1_{ij} - A2_{ij}|}{A1_{ij}} 100\,\% \tag{4}$$

Figure 3 shows the relative difference of the simulations S02 and A09 in (a) and two simulations using A09 in (b) using 10^9 photon packets. It can be seen that the relative difference in (a) is very similar to that in (b), suggesting that the results produced by A09 are statistically similar to that of S02, in both cases smaller than 5 %. We calculated the relative differences between all the simulations obtaining equivalent results (data not shown).

Figure 4 represent the plot of the absorption probability density matrix and the corresponding contours for the simulations using the program A09 with 10^8 photon packets in (a) and 10^{10} in (b). The results are equivalent except for the noise, which is smaller when more photon packets are used. It can be noticed that the absorption probability decreases with depth. This behavior is caused due to the fact that photons are launched at the surface, the propagation of light in the tissue is in the forward direction (the anisotropy factor, $g = 0.8$) and photon packets lose weight as depth increases.

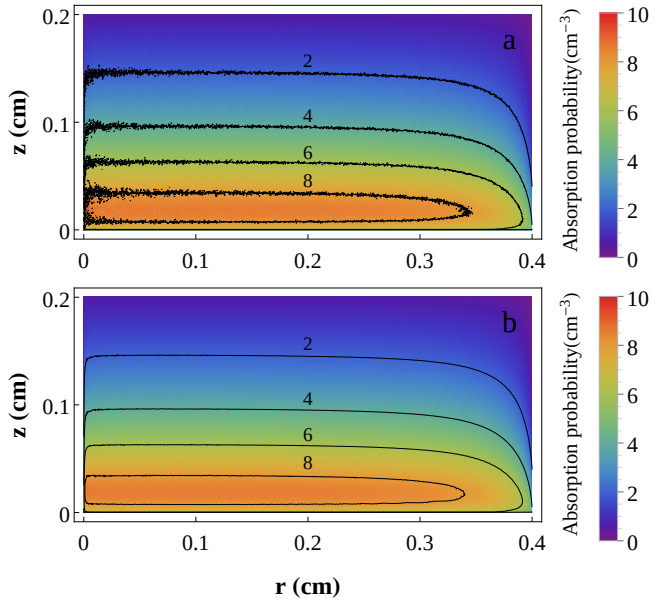

Figure 4. Absorption probability density matrices and the corresponding contours using the program A09. (a) Simulation with 10^8 photon packets.(b) Simulation with 10^{10} photon packets. Both results are equivalent except for the level of noise.

Figure 5 shows how the absorption probability density decreases with depth for two simulations with different number of photon packets. Near the surface the absorption increases with depth due to photon retrodispertion. At a depth of $0.02cm$ the absorption probability density reaches its maximum value and then decreases.

Figure 6 shows the plots of absorption probability density versus radius. Due to the geometry of the model, the absorption probability density is constant for a fixed value of tissue depth z, except in the borders, for radial values near $0.4\,cm$.

Because we do not consider any photon packet outside the tumor tissue, near the borders, the absorbed photon packets arrive mainly from regions with values of radius smaller than the of the cylinder radius representing the tumor tissue. In order to obtain a uniform distribution the incident light source must be larger than this cylinder, but this imply that the number of photons per unit area is smaller and we will need more photon packets in order to obtain the same results.

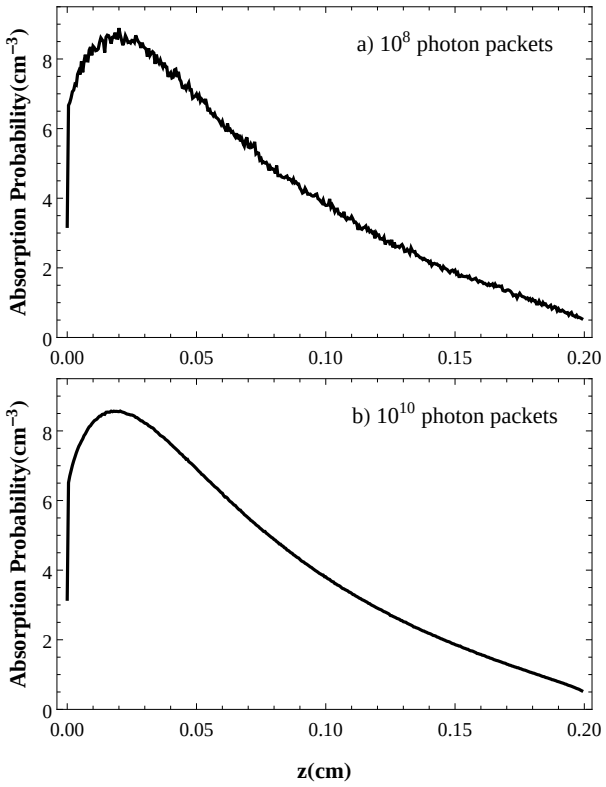

Figure 5. Plot of absorption probability density versus tissue depth for simulations with A09 program at $r = 0.05cm$. (a) Using 10^8 photon packets. (b) Using 10^{10} photon packets.

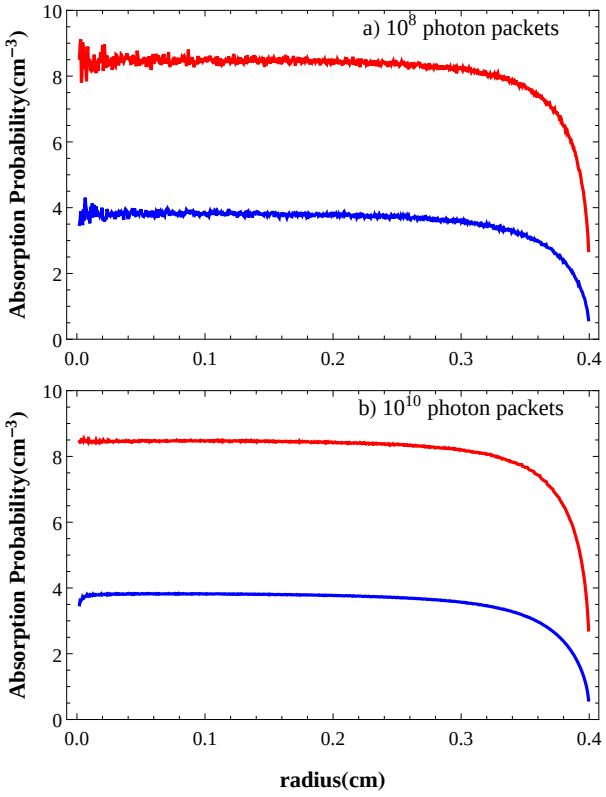

Figure 6. Plot of absorption probability density versus radius at tissue depth of $0.025cm$ in red and $0.1cm$ in blue using the A09 program. (a) For 10^8 photon packets. (b) For 10^{10} photon packets.

V. OPTIMIZATION OF THE EXECUTION TIME

We measured the execution time of every program in order to compare them and determine which one is the optimal. Figure 7 represent the logarithmic plot of the execution time versus the number of photon packets for each implemented code. It shows that the parallel programs are faster than the serial one. In particular A09 is 300 times faster than the serial code (68.9*hours* versus 14*minutes* with 10^{10} photon packets).

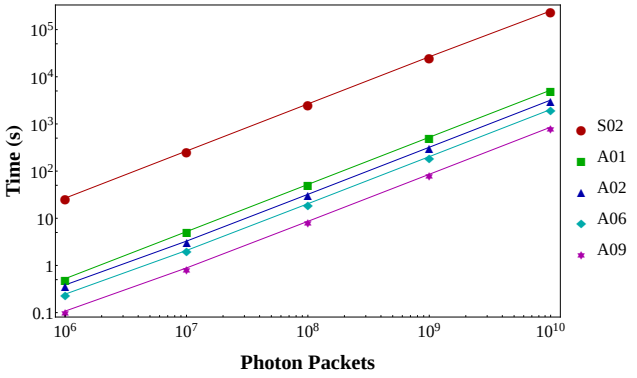

Figure 7. Logarithmic plot of the execution time versus the number of photon packets.

A visual inspection of the plots in figure 7 indicates that the relationship between time, t, and number of photon packets, N, is represented by equation 5. Where C is a constant and m the slope of the curve.

$$t = \frac{N^m}{C} \tag{5}$$

We fitted the data to equation 5 and the results for the calculation of the parameters m and C are shown in Table 2 for all the executed programs. Analyzing the results obtained for m, we can conclude that the relationship between t and N is linear and C is a constant representing the number of photon packets per second calculated by each program. The linear dependency also indicates that C do not depends on N.

Table 2. Parameters obtained after fitting the execution time versus the number of photon packets of each program to equation 5.

Code	m	C (s^{-1})
S02	0.972 ± 0.002	$(2.12 \pm 0.08) \times 10^4$
A01	1.00005 ± 0.00001	$(1.92088 \pm 0.00008) \times 10^6$
A02	0.9996 ± 0.0001	$(3.095 \pm 0.007) \times 10^6$
A06	0.9996 ± 0.0001	$(4.89 \pm 0.01) \times 10^6$
A09	0.9992 ± 0.0001	$(11.51 \pm 0.04) \times 10^6$

VI. CONCLUSIONS

In this work we presented a model to simulate the absorbance probability density function of a monolayer tumoral tissue using the Monte Carlo Method. A serial and four parallel codes where implemented increasing the photon packet number from 10^6 to 10^{10}. We demonstrated that the use of GPU accelerated the execution time 300 times compared with the serial code in a linear way. It is more efficient to use two graphic cards, make the iteration cicles in the GPU and use the intrinsic functions of the ALU. This study corroborate the previous results about the behavior of absorbance probability density with tissue depth and tumor radius. Absorbance decreases with depth and is constant for the same values of radius except in the borders of the model. The Monte Carlo Method is efficient to study the interaction of light with turbid media but requires large execution times. The use of GPU calculation can effectively accelerate this process allowing to increase the number of photon packets obtaining more precise results.

REFERENCES

[1] Allison, R. R. and Moghissi, K. *Photodiagnosis and Photodynamic Therapy* **10**(4), 331–341 (2013).

[2] Zhu, C. and Liu, Q. *Journal of Biomedical Optics* **18**(5), 050902–050902 (2013).

[3] Ishimaru, A. *Applied Optics* **28**(12), 2210 (1989).

[4] Wilson, B. C. and Patterson, M. S. *Physics in Medicine and Biology* **53**(9), R61–109 (2008).

[5] Prahl, S. A., Keijzer, M., Jacques, S. L., and Welch, A. J. *Dosimetry of laser radiation in medicine and biology* **5**, 102–111 (1989).

[6] Wang, L., Jacques, S. L., and Zheng, L. *Computer Methods and Programs in Biomedicine* **47**(2), 131–146 (1995).

[7] Jacques, S. L. and Wang, L. In *Optical-Thermal Response of Laser-Irradiated Tissue*, Welch, A. J. and Gemert, M. J. C. V., editors, Lasers, Photonics, and Electro-Optics, 73–100. Springer US (1995).

[8] Henyey, L. and Greenstein, J. L. *Astrophys. J.* **93**, 70–83 (1941).

[9] Hecht, E. *Optics*. (Addison-Wesley Publishing Company, Inc.), 3 edition, (1987).

[10] Salomatina, E., Jiang, B., Novak, J., and Yaroslavsky, A. N. *J Biomed Opt* **11**(6), 064026 (2006).

[11] Salas-García, I., Fanjul-Vélez, F., and Arce-Diego, J. L. *Optics Communications* **285**(6), 1581–1588 (2012).

[12] Kirk, D. B. and Wen-mei, W. H. *Programming massively parallel processors: a hands-on approach*. Newnes, (2012).

[13] Alerstam, E., Lo, W. C. Y., Han, T. D., Rose, J., Andersson-Engels, S., and Lilge, L. *Biomedical Optics Express* **1**(2), 658–675 (2010).

[14] Marsaglia, G., Tsang, W. W., et al. *Journal of statistical software* **5**(8), 1–7 (2000).