

## PRACTICAL WORK 1

## 1 Introduction

During the practical work sessions, the use of Python to solve the proposed exercises is recommended. Remember to save your work, as it may be used in the next sessions.

A report where you describe your methodology and provided answers for the exercises for all practical work sessions is expected after the end of the course. Send your work to:

[martin.RODRIGUEZ-VEGA@univ-amu.fr](mailto:martin.RODRIGUEZ-VEGA@univ-amu.fr)

Please abstain to use LLM or AI tools. Previous years have shown that false responses are very common. Do not hesitate to ask questions during the session.

## 2 Optical properties for biological tissues

The main parameters we will use to describe the propagation of light in biological tissues are: Absorption coefficient  $\mu_a$ , Scattering coefficient  $\mu_s$ , anisotropy factor  $g$ , and refraction index  $n$ .

We will consider fixed values for the latter three properties,  $\mu_s = 100 \text{ cm}^{-1}$ ,  $g = 0.9$ ,  $n = 1.37$ , and will focus mainly in the changes introduced by modifying the absorption coefficient.

Assume a simplified tissue composed of water, fat, and hemoglobin, with different composition ratios. The absorption coefficient of this tissue can be modeled as a weighted average of its compositional substances[3]:

$$\mu_a^{\text{tissue}}(\lambda) = \sum_{s \in \text{substances}} c_s \mu_a^s(\lambda) \quad (1)$$

where  $\lambda$  is the wavelength,  $c_s \in [0, 1]$  is the percentage of abundance of substance  $s$  in the tissue. Note that  $\sum_s c_s = 1$ .

**Exercise 1.1:** Generate plots of the absorption spectra of water, fat, hemoglobin (Hb), and oxygenated hemoglobin (HbO) in the near infrared range (between 600 and 1000 nm). Which wavelengths would you use to try to determine the blood oxygenation? Why?

Assume a tissue composition of 65% water, 20% fat and 15% total hemoglobin (Hb + HbO). Plot the resulting absorption spectrum of the tissue at for different oxygenation levels. Discuss.

**Hints:** Different spectra for common biological substance can be found in <https://omlc.org/spectra/index.html>. Plot the spectra in log scale (e.g. function `semilogy` of `matplotlib.pyplot` in Python).

To convert hemoglobin molar extinction coefficient [ $\text{cm}^{-1} \text{ M}$ ] into absorption coefficient [ $\text{cm}^{-1}$ ] you can multiply the former by 0.0054 (assuming 15% Hb, and 64.5 molar mass).

## 3 Light propagation model

In highly scattering media,  $\mu_a \ll \mu_s$ , the Diffusion Approximation (DA) is commonly used instead of the Radiative Transfer Equation (RTE)[2]:

$$-\nabla \cdot D(\mathbf{r}) \nabla \phi(\mathbf{r}) + \mu_a(\mathbf{r}) \phi(\mathbf{r}) = s(\mathbf{r}) \quad (2)$$

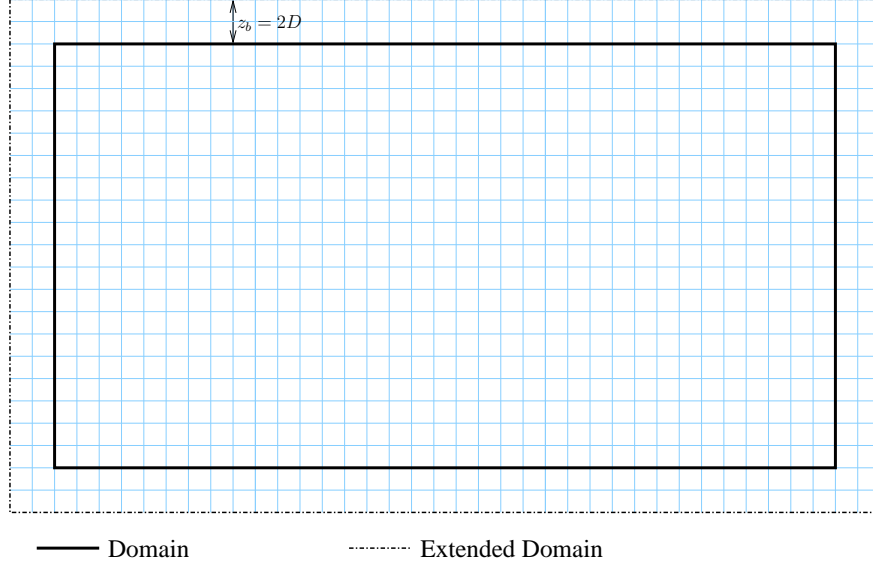


Figure 1: Domain and extended domain to take into account boundary condition

where the diffusion coefficient  $D$  is

$$D(\mathbf{r}) = \frac{1}{3(\mu_a + (1-g)\mu_s)} \quad (3)$$

Common boundary conditions used for the DA include the Extrapolated Boundary Condition (EBC) [1], which are Dirichlet conditions that extend the domain a distance of  $2D$  and impose  $\phi(\mathbf{r}) = 0$  in this new boundary. See Fig. 1.

If we assume that  $D(\mathbf{r})$  is homogeneous (does not change in space), then we can simplify the DA to a modified Helmholtz equation:

$$-\nabla^2 \phi(\mathbf{r}) + k^2(\mathbf{r})\phi(\mathbf{r}) = \frac{s(\mathbf{r})}{D} \quad (4)$$

where  $k^2(\mathbf{r}) = \mu_a(\mathbf{r})/D$ .

### 3.1 Numerical solutions to the DA

We will consider here the Finite Differences Method (FDM), which discretizes space into a grid of finite points,  $\mathbf{r} \in \{\mathbf{r}_0, \mathbf{r}_1, \mathbf{r}_2, \dots, \mathbf{r}_{n-1}\}$ . This method approximates the first and second derivatives using low order Taylor expansions. For instance, in 2D,

$$\begin{aligned} \frac{\partial}{\partial x} \phi(x, y) &\approx \frac{\phi(x + \Delta x, y) - \phi(x, y)}{\Delta x} && \text{Forward difference, 1st derivative} \\ \frac{\partial}{\partial x} \phi(x, y) &\approx \frac{\phi(x, y) - \phi(x - \Delta x, y)}{\Delta x} && \text{Backward difference, 1st derivative} \\ \frac{\partial}{\partial x} \phi(x, y) &\approx \frac{\phi(x + \Delta x, y) - \phi(x - \Delta x, y)}{2\Delta x} && \text{Central difference, 1st derivative} \\ \frac{\partial^2}{\partial x^2} \phi(x, y) &\approx \frac{\phi(x + \Delta x, y) - 2\phi(x, y) + \phi(x - \Delta x, y)}{\Delta x^2} && \text{Central difference, 2nd derivative} \end{aligned} \quad (5)$$

The FDM gives a solution to a differential linear operator by transforming it into a set of linear equations

$$-\nabla^2\phi(\mathbf{r}) + k^2(\mathbf{r})\phi(\mathbf{r}) = \frac{s(\mathbf{r})}{D} \xrightarrow{\text{FDM}} A \begin{bmatrix} \phi(\mathbf{r}_0) \\ \phi(\mathbf{r}_1) \\ \vdots \\ \phi(\mathbf{r}_{n-1}) \end{bmatrix} = \frac{1}{D} \begin{bmatrix} s(\mathbf{r}_0) \\ s(\mathbf{r}_1) \\ \vdots \\ s(\mathbf{r}_{n-1}) \end{bmatrix} \quad (6)$$

where  $A$  is a  $n \times n$  matrix.

**EXERCISE 1.2** : Consider a 1D domain of length  $L$  discretized into  $n$  points with uniform spacing  $\Delta x$ . Write the FDM form of the DA (4) for one point in the interior  $x_i$  whose neighbors are  $x_{i-1}$  and  $x_{i+1}$ . How can you use this expression to generate a matrix?

**Hints** : To simplify notation, you can write  $\phi(x_i) = \phi[i]$  and  $k^2(x_i) = k^2[i]$ . You can think of row  $i$  of matrix  $A$  as the expression you obtain expanding the DA for point  $i$ , and  $A_{ij}$  as the coefficient of  $\phi[j]$  if it appears in the expansion.

**EXERCISE 1.3** : Write a Python function that takes as input a vector of size  $n$  representing the values of  $k^2(x)$ , the discretization spacing  $\Delta x$ , and the vector of sources  $s(x)$  of dimension  $n$ . This function constructs the matrix  $A$  and solves the linear system to return the fluence vector  $\phi(x)$ .

**Hints** : To apply the EBC, note that in 1D they become  $x[0] = 0$  and  $x[n-1] = 0$ . Thus  $A_{0,0} = 1$  and  $A_{0,i} = 0$  for all other  $i$ , and  $A_{n-1,n-1} = 1$  and  $A_{n-1,i} = 0$  for all other  $i$ , with  $s[0] = 0$  and  $s[n-1] = 0$ .

The linear system can be solve in Python using `numpy.linalg.solve`.

**EXERCISE 1.4** : Using the spectra you developed earlier, create an homogeneous slab of length 2 cm and discretize the space into at least 30 points. Generate the optical properties for a wavelength of your choosing and vectors  $k^2(x)$ . Compute the fluence using  $s(x) = (1 \text{ W cm}^{-2})\delta(x - 1 \text{ cm})$  and plot your results.

Compare your results with the analytical solution of the Helmholtz equation.

**Hints** : Be careful on the discretization of  $\delta$  sources. For some function  $g(x) = p\delta(x - x_i)$ , then  $g[i] = p/\Delta x$  and 0 else.

The Green's function of the Helmholtz equation for an infinite homogeneous medium:

$$G(x, x') = \frac{e^{-k|x-x'|}}{2kD} \quad (7)$$

where  $x'$  is the position of a point source. Recall that the solution for a general source using the Green's function is

$$\phi(x) = \int_0^L G(x, x')s(x')dx' \quad (8)$$

For simple finite geometries, you can use the method of images.

**EXERCISE 1.5** : Try your FDM code for different source positions and non-homogeneous  $k^2$  values.

**EXERCISE 1.6** : Consider a 2D rectangular domain of dimensions  $L_x \times L_y$  discretized into  $n_x \times n_y$  points with uniform spacing. Write the FDM form of the DA (4) for one point in the interior  $\mathbf{r}_i = (x_i, y_i)$ . How many neighbors do each interior point have? How can you use this expression to generate a matrix?

**EXERCISE 1.7** Write a Python function that takes as input a matrix  $k^2[i, j]$ , a matrix  $s[i, j]$  and the discretization step  $\Delta x$ . This function returns the approximate solution to the DA using the FDM.

**Hints** : Note the unknown function  $\phi$  now is 2D matrix  $\phi[i, j]$  (How many unknowns are there?). A 2D array (in fact and n-dimensional array) can be written in 1D, this is called a *ravel*. This is usually done by unwrapping the matrix row by row:

$$\text{vec}(\phi) = (\phi[0, 0], \phi[0, 1], \phi[0, 2], \dots, \phi[0, n_y - 1], \phi[1, 0], \phi[1, 1], \dots, \phi[1, n_y - 1], \dots, \phi[n_x - 1, n_y - 1]) \quad (9)$$

In Python, see the documentation of `numpy` methods `ravel`, `flatten`, `ravel_multi_index`, `unravel_index`.

## References

- [1] R. C. Haskell, L. O. Svaasand, T.-T. Tsay, T.-C. Feng, B. J. Tromberg, and M. S. McAdams. Boundary conditions for the diffusion equation in radiative transfer. *Journal of the Optical Society of America A*, 11(10):2727, 1994.
- [2] A. Ishimaru. *Wave propagation and scattering in random media*, volume I. Academic Press, 1978.
- [3] S. L. Jacques. Optical properties of biological tissues: A review. *Physics in Medicine and Biology*, 58(11), 2013.