NON LINEAR INVERSE PROBLEMS | MScT CSE
Martin Rodriguez-Vega | Year: 2025–2026

PRACTICAL WORK 3

1 Instructions

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.

Documents and useful scripts can be found in the course repository:

https://github.com/m93rodriguez/BioMed-Inversion-Course

The final report report should include:

- Answers to the Exercises proposed in each of the three Practical Work sessions. Mark which exercise you are answering (*e.g.* E1.3, E2.5, E3.1, etc). Organize your report into sections 1 to 3, according to each practical session.
- A description of the methodology of what you implemented and how.
- When showing the results of simulations for which you decided the parameters, clearly indicate the values of each relevant parameter you used, **with units**. Another person should be able to replicate your results using your report.
- Figures showing reconstructions and measurements should also include the original object when applicable. Recall that all figures should be **readable, clear** and **well-labeled**.
- If a plot has values that range over several orders of magnitude, consider using a logarithmic plot. For instance, if showing the convergence of an algorithm and iteration 1 has error 10^4 and after iteration 20 error are below 1, this cannot be clearly seen in a linear plot, and log plot should be used.

Send your work to:

martin.RODRIGUEZ-VEGA@univ-amu.fr

mark the subject as **[BioMed] TP Report - YOUR NAME**. Deadline is:

Monday, February 2nd 2026 at 8h00

I will confirm reception of your email, so try sending me another email if you have not received my reply on Monday afternoon.

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 Diffuse Optical Tomography

In this session, we will use the forward model given by the Diffusion Approximation (DA):

$$-\nabla^2\phi(\mathbf{r}) + k^2(\mathbf{r})\phi(\mathbf{r}) = \frac{s(\mathbf{r})}{D} \tag{1}$$

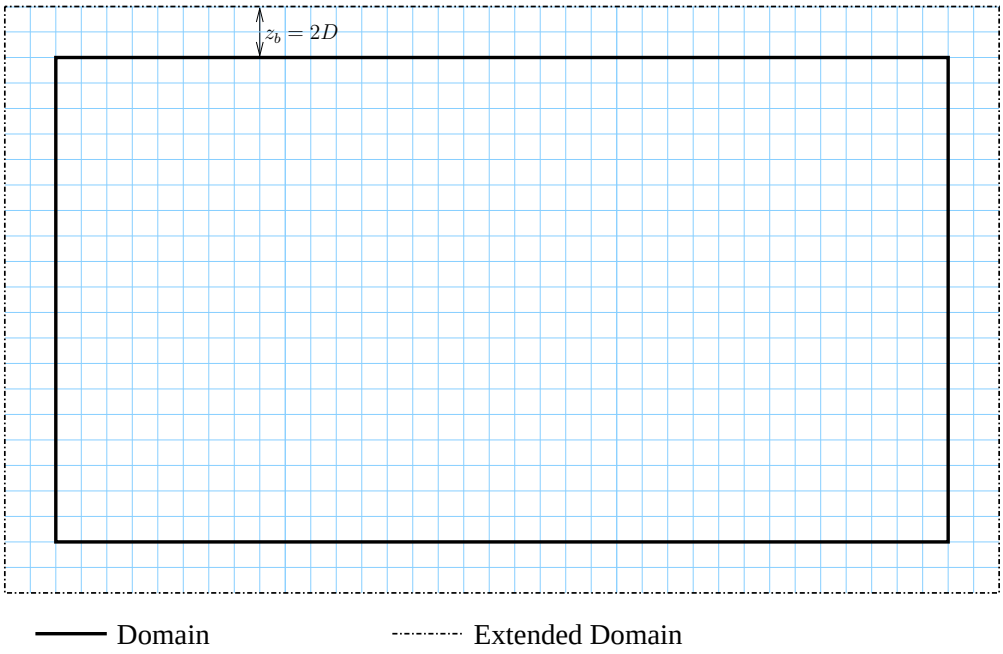

Figure 1: Domain and extended domain to take into account boundary conditions.

where we assume the diffusion coefficient $D \approx 0.4$ mm everywhere. We will use Extended Boundary conditions, where we extend the domain a distance $2D$ in every direction imposing $\phi(\mathbf{r}) = 0$ for $\mathbf{r}$ in the extended boundary, see Fig. 1.

We will use a two-dimensional rectangular extended domain of 10 mm $\times$ 15 mm, with inhomogeneous properties $k^2(\mathbf{r}) \in [0.01, 0.1]$ mm^{-2}. The use of discretization steps Δ of 0.1 mm or 0.2 mm is recommended.

The sources used in this session will be point-like sources $s(\mathbf{r}) = P\delta(\mathbf{r} - \mathbf{r}_s)$, where $P = 2.5$ W/mm such that the right hand side of (1) simplifies to 1.

Attention: In the repository under the `Scripts` directory, you will find Python module `diffapprox.py` which contains method `da2d` that solves the two-dimensional DA using the Finite Element method. You can use this function to solve the forward problem, or you can use your own software from Practical Session 1.

2.1 Measurements

We will consider a set of sources $\mathbf{r}_s \in \mathcal{S}$ and a set of detectors $\mathbf{r}_d \in \mathcal{D}$. We will consider as measurements the values of the fluence for every source-detector combination $\phi(\mathbf{r}_d|\mathbf{r}_s)$, where this reads *the fluence at point* $\mathbf{r}_d$ *given source* $\mathbf{r}_s$.

EXERCISE 3.1 : Generate a domain with uniform properties $k^2(\mathbf{r}) = 0.05$ mm^{-2}. Distribute 10 sources along the horizontal axis at the top, and 15 detectors along the horizontal axis at the bottom. How many measurements can you obtain? Show the resulting measurements.

HINTS : Remember to leave a margin of $2D$ from the borders of the domain, no source or detector should be put there. Plot the measurements as a matrix.

EXERCISE 3.2 : Modify the $k^2(\mathbf{r})$ of the previous domain to add inhomogeneities (regions of more and of less absorption). For the same source/detector positions, show the measurements of the inhomogeneous

domain.

EXERCISE 3.3 : Compare the measurements from the previous two exercises, both by subtraction and division. Discuss what this means for the inverse problem in terms of stability and conditioning.

2.2 Sensitivity matrix and Born Approximation

For the inverse problem, we want to compute how much does changes δk^2 in the parameters k^2 will generate changes in the measurements $\delta\phi(\mathbf{r}_d|\mathbf{r}_s)$. The Born approximation is

$$\delta\phi(\mathbf{r}_d|\mathbf{r}_s) \approx -\int_\Omega G(\mathbf{r},\mathbf{r}_d)\phi(\mathbf{r}|\mathbf{r}_s)\delta k^2(\mathbf{r})\mathrm{d}\mathbf{r} \tag{2}$$

where $G(\mathbf{r},\mathbf{r}_d)$ is the Green's function of the DA for a source placed at $\mathbf{r}_d$.

We can discretize this equation in the domain Ω as $\delta\boldsymbol{\phi} = K\mathbf{x}$, where $\delta\boldsymbol{\phi}$ is the vector of measurements variations, $\mathbf{x}$ is the vector of parameters variations, and K is the sensitivity matrix.

EXERCISE 3.4 : What are the dimensions of $\mathbf{x}$, $\delta\boldsymbol{\phi}$ and J, in terms of the number of sources, detectors, and domain shape?

To improve the conditioning of the inverse problem, instead of considering the absolute change $\delta\phi$ in the measures, it is better to consider the relative change:

$$\frac{\delta\phi(\mathbf{r}_d|\mathbf{r}_s)}{\phi(\mathbf{r}_d|\mathbf{r}_s)} \approx -\frac{1}{\phi(\mathbf{r}_d|\mathbf{r}_s)}\int_\Omega G(\mathbf{r},\mathbf{r}_d)\phi(\mathbf{r}|\mathbf{r}_s)\delta k^2(\mathbf{r})\mathrm{d}\mathbf{r} \tag{3}$$

which is equally discretized as $\mathbf{y} = J\mathbf{x}$, where $\mathbf{y}$ is the vector of relative measure variation and J is the relative sensitivity. This is known as the Rytov Approximation.

EXERCISE 3.5 : Program a function that computes matrix J for a given domain, sources, and detectors. Compute J for the inhomogeneous domain you made in E3.2.

2.3 Inversion

The Gauss-Newton algorithm is used to solve non-linear optimization problems

$$\min_{k^2(\mathbf{r})} \quad \sum_{\mathbf{r}_s,\mathbf{r}_d} |\phi_{\text{real}}(\mathbf{r}_d|\mathbf{r}_s) - \phi(\mathbf{r}_d|\mathbf{r}_s,k^2)|^2 \tag{4}$$

The algorithm starts from a starting "guess" $k_0^2(\mathbf{r})$, which we will iteratively adjust

$$k_{n+1}^2(\mathbf{r}) = k_n^2(\mathbf{r}) + \Delta k^2(\mathbf{r}) \tag{5}$$

where $\Delta k^2(\mathbf{r})$ solves the linear least-squares problem

$$\Delta k^2(\mathbf{r}) = \underset{\mathbf{x}}{\operatorname{argmin}} \quad ||\mathbf{y} - J\mathbf{x}||^2 \tag{6}$$

where we have used the Rytov Approximation. *Note:* the Born approximation can also be used, but it generally converges more slowly, though it can be more stable.

EXERCISE 3.6 : Make a function that implements one iteration of the Gauss-Newton algorithm; it: computes ϕ using the given k^2, computes the measurement error vector $(\phi_{\text{real}} - \phi)/\phi$, computes the relative sensitivity, solves the linear least squares problem, and updates the parameters k^2.

Test this to reconstruct your inhomogeneous properties from E3.2 starting from the domain from E3.1.

HINTS : To solve the linear least squares, you can use the iterative solvers from Practical Session 2.

EXERCISE 3.7 : In the `Data` directory in the repository, you will find a numpy file `dot_data.npz` that contains arrays `sources, detectors, measures`:

- Source coordinates are given as a $S \times 2$ array, where the first column gives the coordinates along the vertical dimension, and the second column gives the coordinates along the horizontal dimension.
- Detector coordinates are given as a $D \times 2$ array, with the same convention as the sources.
- The measurement data is given as a $S \times D$ array, with the fluence $\phi(\mathbf{r}_d|\mathbf{r}_s)$

Use the Newton-Gauss method to estimate the real k^2 parameters, and show your results.

HINTS : The simulated domain was of dimensions 10 mm $\times$ 15 mm (vertical $\times$ horizontal), discretized into 50$\times$75 elements ($\Delta x = 0.2$ mm). Use a starting homogeneous guess of $k^2 = 0.05$ mm^{-2}.

The real parameters are bounded inside $(0.01, 0.1)$ mm^{-2}, so you can impose these bounds in your iterative process.